Assignment 7: Time Series Analysis

Kathlyn MacDonald

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 Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

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

```
#1
getwd()
```

```
## [1] "C:/Users/kmac9/OneDrive/Documents/Duke/Year 1/Spring 2022/ENV Data/Environmental_Data_Analytics_
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("lubridate")
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.3

##
## Attaching package: 'lubridate'

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

```
#install.packages("zoo")
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
#install.packages("trend")
library(trend)
library(dplyr)

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

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.

```
#2
EPAir_O3_2010 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
EPAir_O3_2011 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
EPAir_O3_2012 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
EPAir_O3_2013 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
EPAir_O3_2014 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
EPAir_O3_2015 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
EPAir_O3_2016 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
EPAir_O3_2017 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
EPAir_O3_2018 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
EPAir_O3_2019 <-read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(EPAir_O3_2010,EPAir_O3_2011,EPAir_O3_2012,EPAir_O3_2013,EPAir_O3_2014,EPAir_O3_20
```

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
GaringerOzoneWrangled <-select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALU

# 5

Days <-as.data.frame(seq(as.Date('2010-01-01'), as.Date('2019-12-31'), by = 'days'))
colnames(Days) <- c("Date")

# 6

GaringerOzone <- left_join(Days, GaringerOzoneWrangled, by = "Date")
```

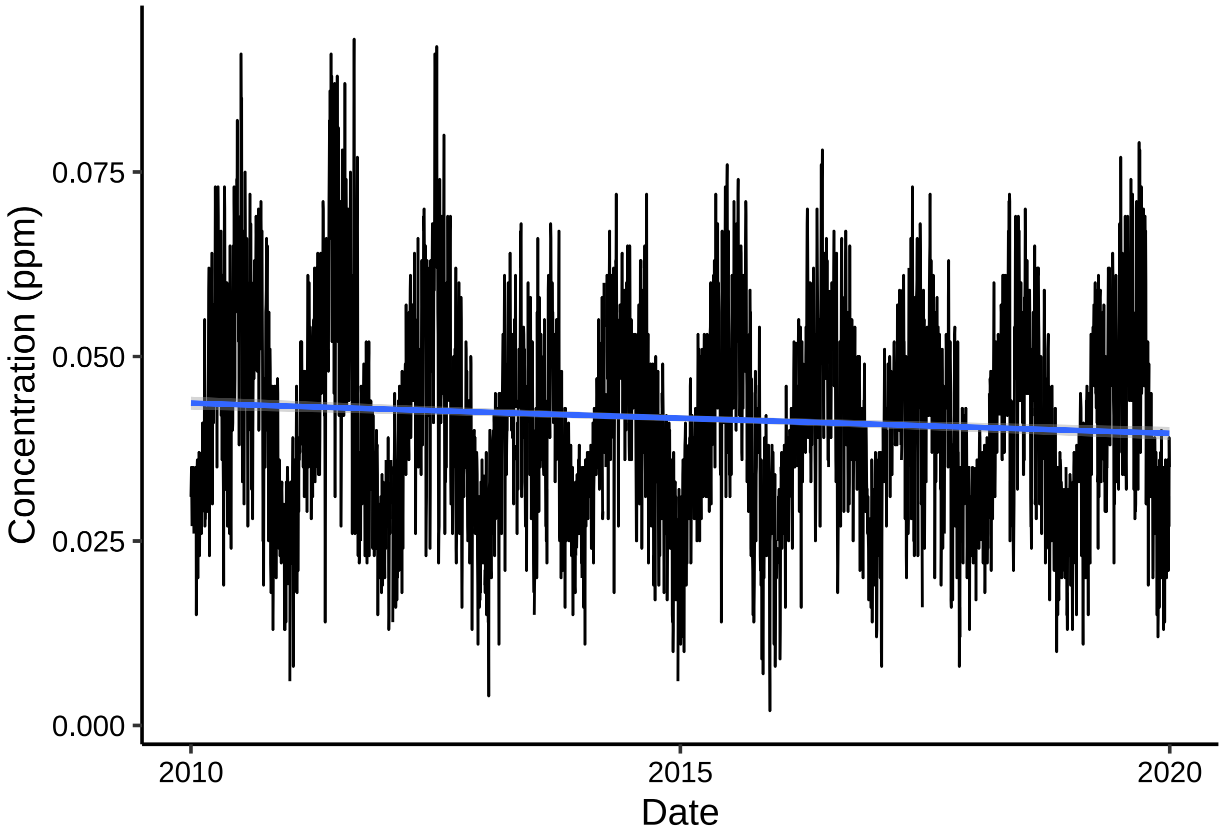
Visualize

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

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  labs(x= "Date", y = "Concentration (ppm)") +
  geom_smooth(method = 'lm')
```

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

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



Answer: Ozone levels while they fluctuate seasonally, have remained relatively constant. However, it appears that there has been a slight decline in ozone concentration over the decade.

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$Daily.Max.8.hour.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
```

```
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati
```

Answer: We didn't use piecewise constant because the data is seasonal, so choosing the 'nearest neighbor' would result in different answers depending on if you chose a point ahead or behind. Additionally, we didn't use a spline interpolation because the data isn't quadratic.

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 <- GaringerOzone %>%
  mutate(Year = year(Date)) %>%
  mutate(Month = month(Date)) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  dplyr:: group_by(Date, Month, Year) %>%
  dplyr:: summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration.clean)) %>%
  select(Date, MeanOzone)
```

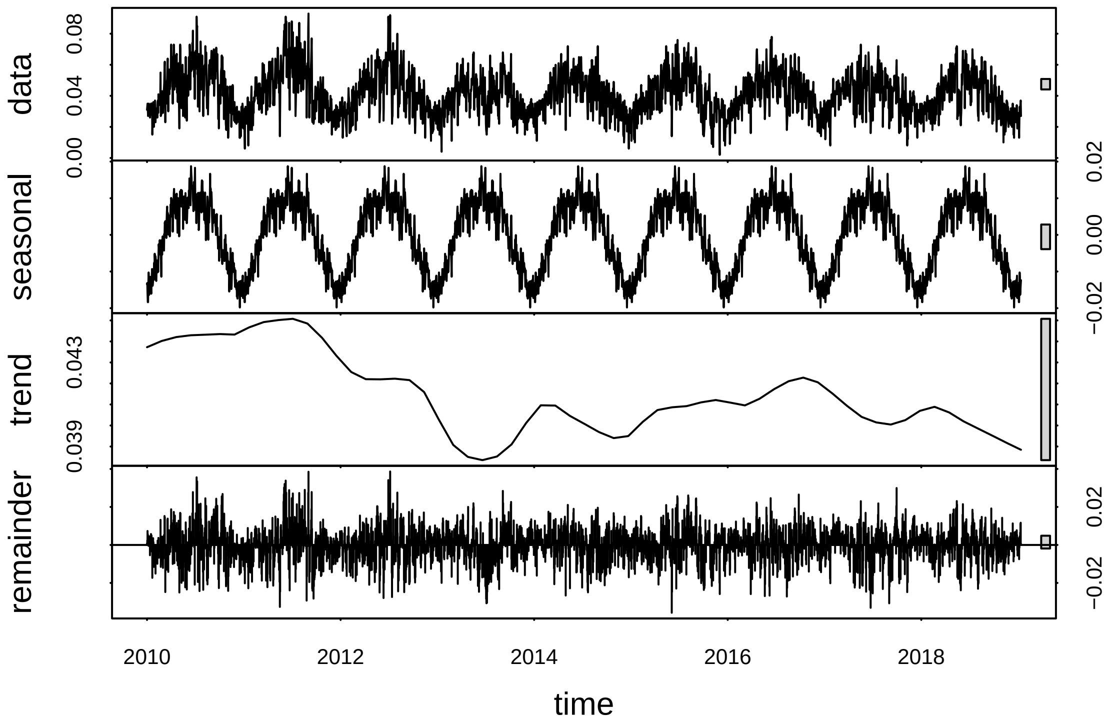
```
## ‘summarise()‘ has grouped output by ’Date’, ’Month’. You can override using the ‘.groups‘ argument.

## Adding missing grouping variables: ‘Month‘
```

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.

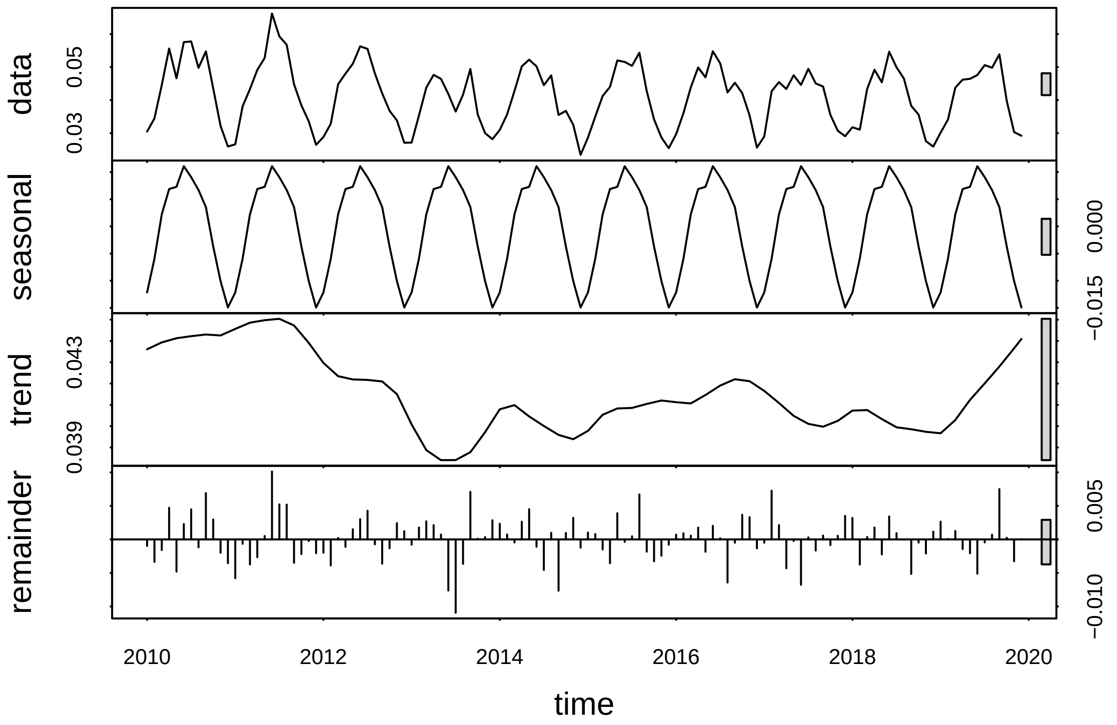
```
#10

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration.clean, start = c(2010,1

GaringerOzone.monthly.ts <-ts(GaringerOzone.monthly$MeanOzone, start = c(2010,1), end = c(2019,12), fre
```

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")

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
Ozone_data_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

Ozone_data_trend
```

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

```
summary(Ozone_data_trend)
```

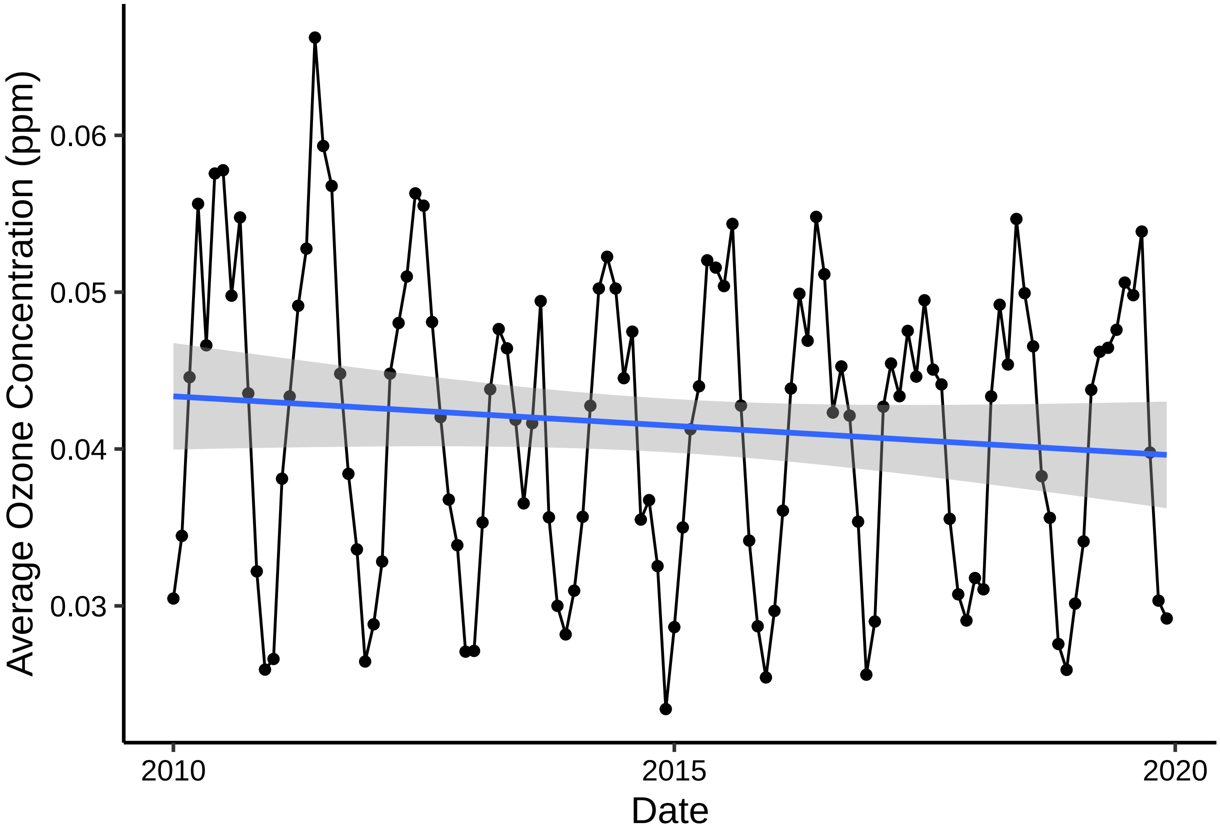
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal mann kendall test is the only appropriate test because the ozone concentration clearly fluctuates with seasons, and this test is the only one that takes this into account.

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

Ozone_concentration_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = MeanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Ozone Concentration (ppm)") +
  geom_smooth( method = lm )
print(Ozone_concentration_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: We were testing to see if ozone concentration has changed over the course of measurement, while maintaining the seasonal component. In this case we reject the null hypothesis and conclude that there is a difference between the means (p = 0.0467). While this is value is technically significant (p<0.05), it is close enough to be questionable.

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 and turn them into data frames

Ozone_concentration_components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

Ozone_concentration_components <- mutate(Ozone_concentration_components,
        Observed = GaringerOzone.monthly$MeanOzone,
        Date = GaringerOzone.monthly$Date)

#16
Ozone_concentration_components_ts <- ts(Ozone_concentration_components$trend,
                   start=c(2010,1),
                   frequency=12)
Ozone_concentration_components_trend <- Kendall::MannKendall(Ozone_concentration_components_ts)
summary(Ozone_concentration_components_trend)
```

```
## Score =  -1922 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

Answer: After seasonality has been removed from the data set we were still able to reject the null hypothesis, and conclude that there is a difference between the means across the measurement period (p = 1.3168e-05). The significance of the data set is now small enough, that the significance is no longer questionable.