Assignment 8: Time Series Analysis

Kathleen Mason

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., "Salk_A06_GLMs_Week1.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 3 at 1:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme
- 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). Call these GaringerOzone201*, with the star filled in with the appropriate year in each of ten cases.

```
getwd()
```

```
## [1] "/Users/kathleenmason/Documents/DUKE/Data Analytics/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
library(lubridate)
library(trend)
library(zoo)
library(dplyr)

GaringerOzone2012010<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
GaringerOzone2012011<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
GaringerOzone2012012<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
GaringerOzone2012013<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
GaringerOzone2012014<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
GaringerOzone2012015<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
GaringerOzone2012016<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
GaringerOzone2012017<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
```

```
GaringerOzone2012018<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
GaringerOzone2012019<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")
```

Wrangle

2. Combine your ten datasets into one dataset called GaringerOzone. Think about whether you should use a join or a row bind.

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.

```
# 2
GaringerOzone.10<-rbind(GaringerOzone2012010, GaringerOzone2012011, GaringerOzone2012012, GaringerOzone2



# 3
GaringerOzone.10$Date<- as.Date(GaringerOzone.10$Date, format = "%m/%d/%Y")

# 4
GaringerOzone.Wrangled<- select(GaringerOzone.10, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_

# 5
Days <- as.data.frame( seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 day"))
names(Days)[1] <- "Date"

# 6
#specify 3652 rows and 3 columns.
#left_join(x,y)
GaringerOzone<- left_join(Days, GaringerOzone.Wrangled)
```

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

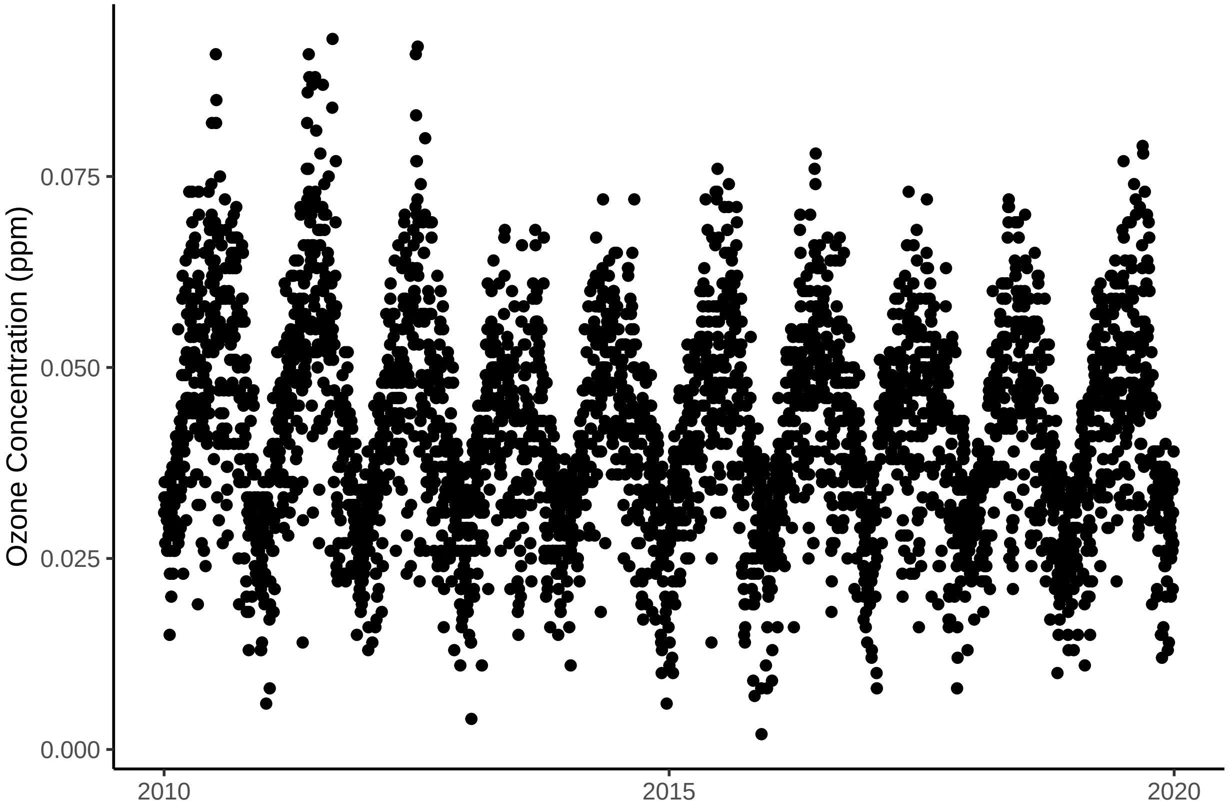
```
#put days first because Rows in x with no match in y will have NA values in the new columns.
```

Visualize

7. Create a ggplot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly.

```
ggplot(GaringerOzone)+
  geom_point(aes(x= Date,
                 y= Daily.Max.8.hour.Ozone.Concentration))+
  theme_classic()+
  ylab("Ozone Concentration (ppm)")+
  xlab("")
```

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



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?

   Answer: In a piecewise interpolation, the nearest neighbor approach is used for missing data, which creates a step type calculation of the NA values. A linear interpolation will allows a straight line between values with a missing value in between, this creates a smoother approach to interpolating with missing values.

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)

10. Generate a time series called GaringerOzone.monthly.ts, with a monthly frequency that specifies the correct start and end dates.

11. Run a time series analysis. In this case the seasonal Mann-Kendall is most appropriate; why is this?

    Answer: we have identical distribution, but assumes no temporal autocorrelation. However, our data is daily and might have temporal autocorrelation so we bring it down to monthly. The Mann-Kendall test is the only one that allows seasonality, and from our plot we can see there might be some seasonality.

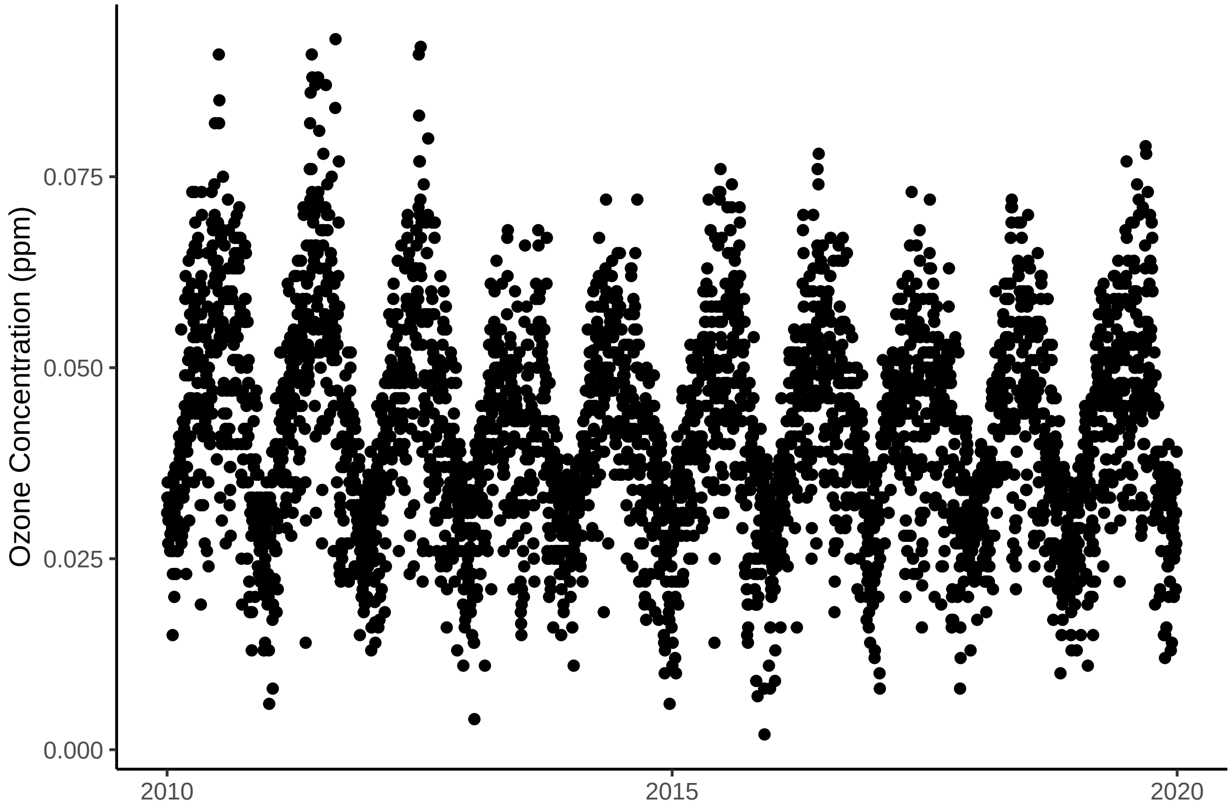
12. To figure out the slope of the trend, run the function `sea.sens.slope` on the time series dataset.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. No need to add a line for the seasonal Sen's slope; this is difficult to apply to a graph with time as the x axis. Edit your axis labels accordingly.

```
# 8
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co

GaringerOzone.plot <-
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point(aes(x= Date,
                 y= Daily.Max.8.hour.Ozone.Concentration)) +
               theme_classic()+
  ylab("Ozone Concentration (ppm)")+
  xlab("")

print(GaringerOzone.plot)
```



```
# 9
GaringerOzone.Monthly<- GaringerOzone %>%
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone.Monthly$Date <- as.Date(paste(GaringerOzone.Monthly$Year,GaringerOzone.Monthly$Month,
                                             1, sep="-"),
                                       format = "%Y-%m-%d")
```

```r
# 10
#Generate a time series called GaringerOzone.monthly.ts, with a monthly frequency that specifies the co
GaringerOzone.Monthly.ts<- ts(GaringerOzone.Monthly$MeanOzone, frequency = 12, start = c(2010, 01, 01),


# 11
#Run a time series analysis. In this case the seasonal Mann-Kendall is most appropriate; why is this?
GaringerOzone.Monthly.ts.trend <- smk.test(GaringerOzone.Monthly.ts)
GaringerOzone.Monthly.ts.trend
```

```
##
##  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
```

```r
summary(GaringerOzone.Monthly.ts.trend)
```

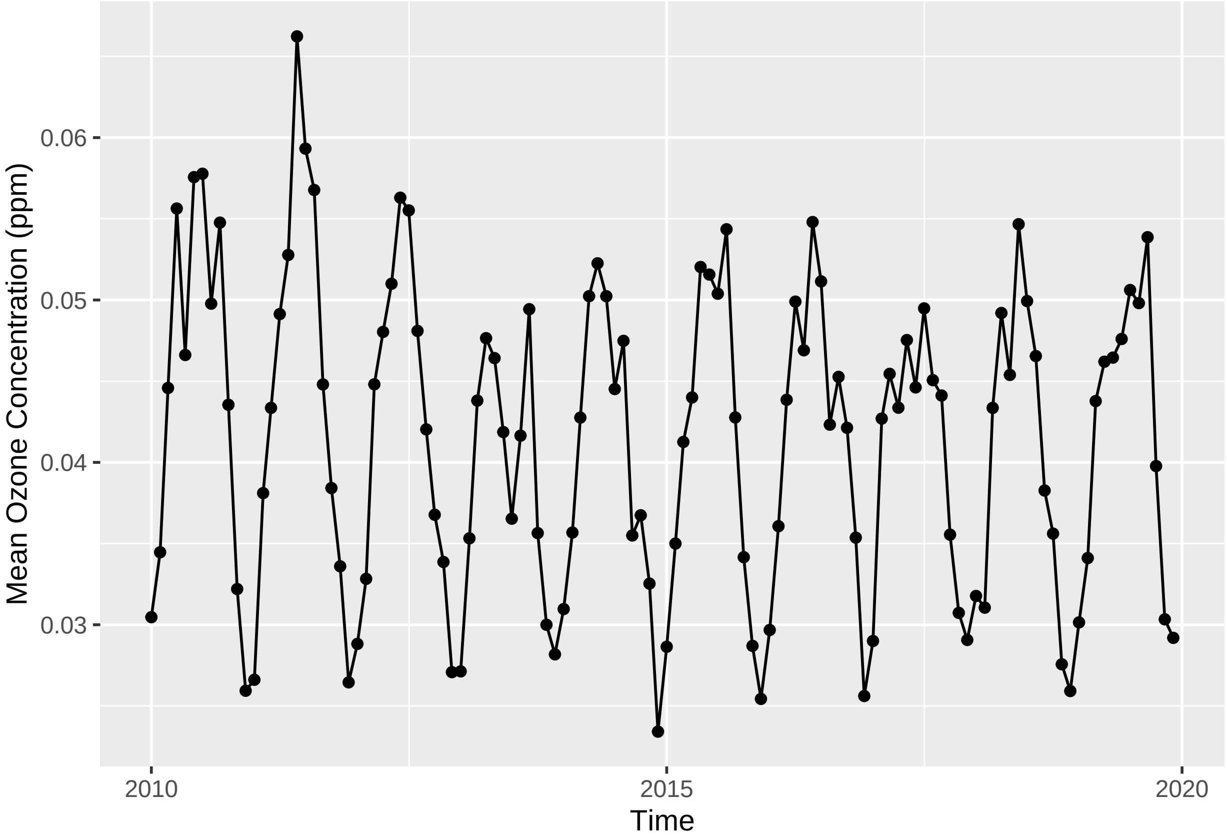
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.Monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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
```

```r
# 12
#To figure out the slope of the trend, run the function `sea.sens.slope` on the time series dataset.
sea.sens.slope(GaringerOzone.Monthly.ts)
```

```
## [1] -0.0002044163
```

```r
#slope = -0.0002044
```

```
# 13
#Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom
GaringerOzone.Monthly.Plot <-
ggplot(GaringerOzone.Monthly, aes(x = Date, y = MeanOzone)) +
  geom_point() +
  geom_line()+
  ylab("Mean Ozone Concentration (ppm)")+
  xlab("Time")
print(GaringerOzone.Monthly.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: Mean Ozone Concentration does follow a monotonic trend. There is a significant monthly trend from 2010 to 2019, and we did not detect any individual trends between months. (seasonal Mann-Kendall, pvalue <0.05, Z= -1.963)

    pvalue less than 0.05 means significant trend, because it can't happen randomly,it must be following a trend.