Assignment 8: Time Series Analysis

Sierra Kindley

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.

```
# 1
getwd() #verify working directory
```

```
## [1] "C:/Users/sierr/Documents/Duke University/ENVIRON-872L/Environmental_Data_Analytics_2020"
```

```
library(tidyverse) #load tidyverse packages
library(lubridate) #load lubridate package
library(zoo) #load zoo package
library(trend) #load trend package

sierratheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
  #build new theme 'sierratheme' and define its parameters
theme_set(sierratheme) #set sierratheme as default plot theme

GaringerOzone2010 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
GaringerOzone2011 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
```

```
GaringerOzone2012 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
GaringerOzone2013 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
GaringerOzone2014 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
GaringerOzone2015 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
GaringerOzone2016 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
GaringerOzone2017 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
GaringerOzone2018 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
GaringerOzone2019 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")
#import the ten Garinger High School raw ozone time series datasets
```

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.combined <-
  rbind(GaringerOzone2010, GaringerOzone2011, GaringerOzone2012, GaringerOzone2013,
            GaringerOzone2014, GaringerOzone2015, GaringerOzone2016, GaringerOzone2017,
            GaringerOzone2018, GaringerOzone2019)
  #combine ten ozone time series datasets into single dataset

# 3
GaringerOzone.combined$Date <-
  as.Date(GaringerOzone.combined$Date, format = "%m/%d/%Y")
  #format 'Date' column as date
class(GaringerOzone.combined$Date) #verify class of 'Date' column
```

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

```
# 4
GaringerOzone.subset <-
  select(GaringerOzone.combined, Date, Daily.Max.8.hour.Ozone.Concentration,
          DAILY_AQI_VALUE)
  #Remove columns that are not of interest for this analysis
```

```
# 5
Days <-
  as.data.frame(colname = "Date", seq(as.Date("2010-01-01"),
                                    as.Date("2019-12-31"), by = "day"))
  #create new data frame with desired sequence of dates
names(Days)[1] <- paste("Date")
  #name column in new data frame 'Date'


# 6
GaringerOzone <-
  left_join(Days, GaringerOzone.subset)
```

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

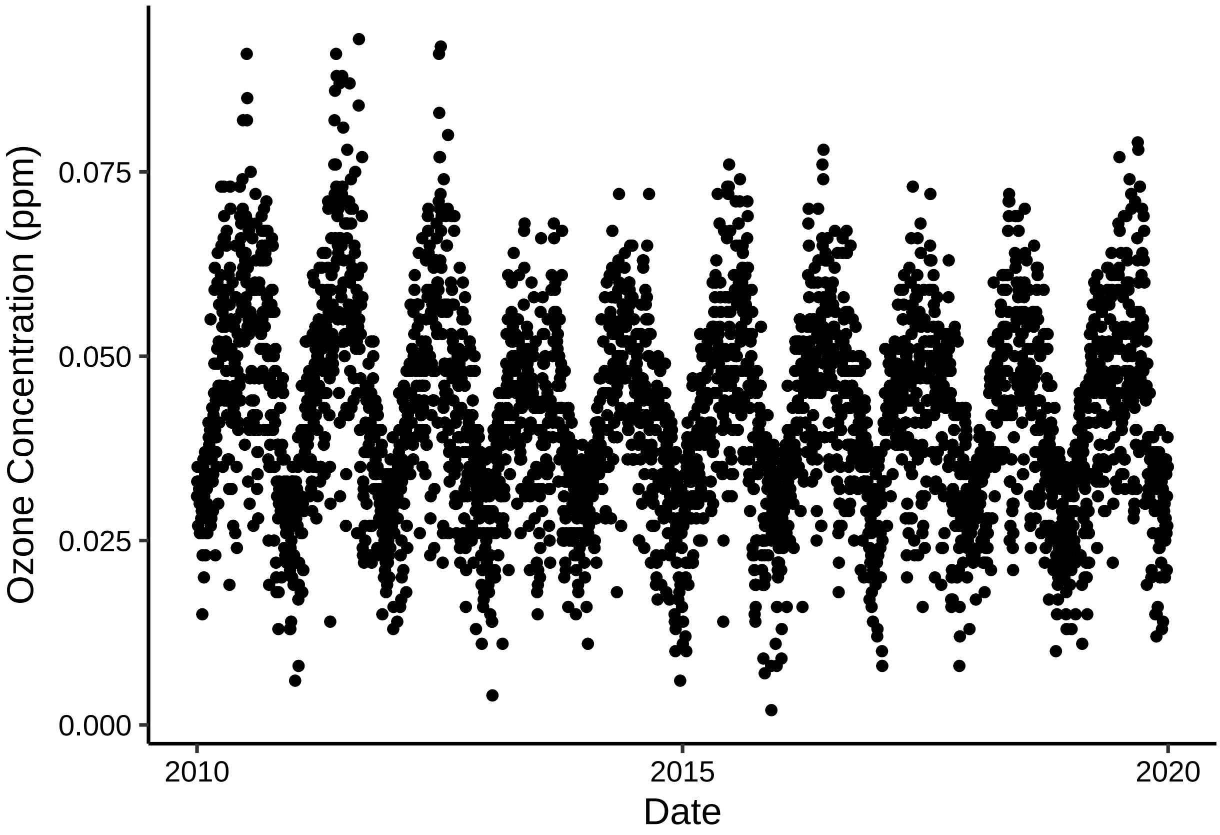
```
  #combine the 'Days' and 'GaringerOzone.subset' data frames
```

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.

```
# 7
ozone.plot1 <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_point() +
    labs(x = expression("Date"), y = expression("Ozone Concentration (ppm)"))
print(ozone.plot1)
```

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



```
#define plot parameters and aesthetics, and print plot
```

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: A linear interpolation is most appropriate to use here, as it is a "connect the dots" approach. Missing data are assumed to fall between the previous and subsequent measurement, and a straight line is drawn between the known points. Thus, the values of the interpolated data on any given date are determined. As we do not expect the data to vary/change by a large amount from point to point and can reasonably assume that each value is likely to fall somewhere between the values before and after it, this type of interpolation is appropriate to use.

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: Here, the seasonal Mann-Kendall test is most appropriate, as we need to account for seasonality in our data, and it allows us to do so. The other tests do not. In addition, the seasonal Mann-Kendall test does not make assumptions about the distribution of our data and does not require that our data be normally distributed (it's a non-parametric test). The seasonal Mann-Kendall test does not autocorrelate values at a specific time with prior or successive values (no temporal autocorelation), which is also desirable in this case.

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.Concentration)
  #fill in missing daily ozone data using linear interpolation

# 9
GaringerOzone.monthly <- #create new data frame
  GaringerOzone %>% #pipe from 'GaringerOzone' dataset
    mutate(Year = year(Date),
            Month = month(Date)) %>% #add columns for month and year
    group_by(Year, Month) %>% #group data by year and month
    summarise(Mean_Ozone_Concentration = mean(Daily.Max.8.hour.Ozone.Concentration))
      #generate mean ozone concentrations for each month

GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$Year,
                                            GaringerOzone.monthly$Month,
                                                1, sep = "-"),
                                         format = "%Y-%m-%d")
  #create new 'Date' column with each month-year combo. set as first day of month

# 10
GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Mean_Ozone_Concentration, frequency = 12,
      start = c(2010, 1, 1), end = c(2019, 12, 1))
  #generate new time series and define monthly frequency

# 11
GaringerOzone.monthly.smk <-
  smk.test(GaringerOzone.monthly.ts)
  #run SMK time series analysis

GaringerOzone.monthly.smk
```

```
##
##  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.smk)
```

```
##
```

```
##  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
```

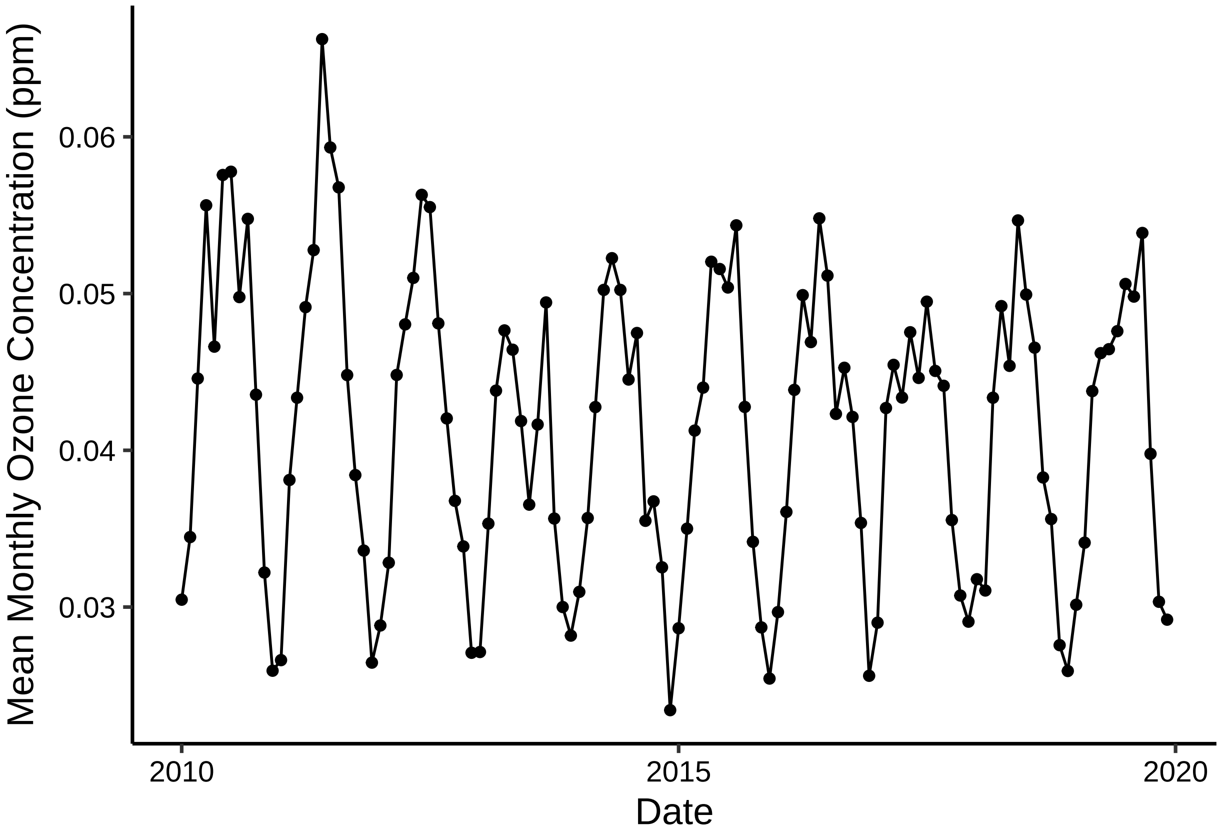
```r
  #generate summary of SMK time series analysis results

# 12
sea.sens.slope(GaringerOzone.monthly.ts)
```

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

```r
  #find the slope of the trend from the time series analysis

# 13
ozone.plot2 <-
  ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone_Concentration)) +
    geom_point() +
    geom_line() +
    labs(x = expression("Date"), y = expression("Mean Monthly Ozone Concentration (ppm)"))
print(ozone.plot2)
```



```
#define plot parameters and aesthetics, and print 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: Based on data from 2010-2019, mean monthly ozone concentrations at Garinger High School in North Carolina have a significant seasonal trend (SMK, p-value < 0.05, z = -1.963). This significant trend is best quanified by a Seasonal Sen's Slope of -0.000204. Please note that the p-value resulting from the seasonal Mann-Kendall test was very close to 0.05, but was less than 0.05 nonetheless.