Assignment 8: Time Series Analysis

Rachel Gonsenhauser

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/rachelgonsenhauser/Documents/Environmental_Data_Analytics_2020"
```

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

mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
theme_set(mytheme)

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")
```

```r
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")
```

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 2019-12-13 (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 comine 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.

```r
# 2
GaringerOzone <- rbind(GaringerOzone2010, GaringerOzone2011, GaringerOzone2012, GaringerOzone2013, Garin

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

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

```r
# 4
GaringerOzone.wrangled <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <-
  as.data.frame(seq(as.Date('2010-01-01'),
                    as.Date('2019-12-31'), by = 'days'), times = num_repeat)
colnames(Days) <- c("Date")

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

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

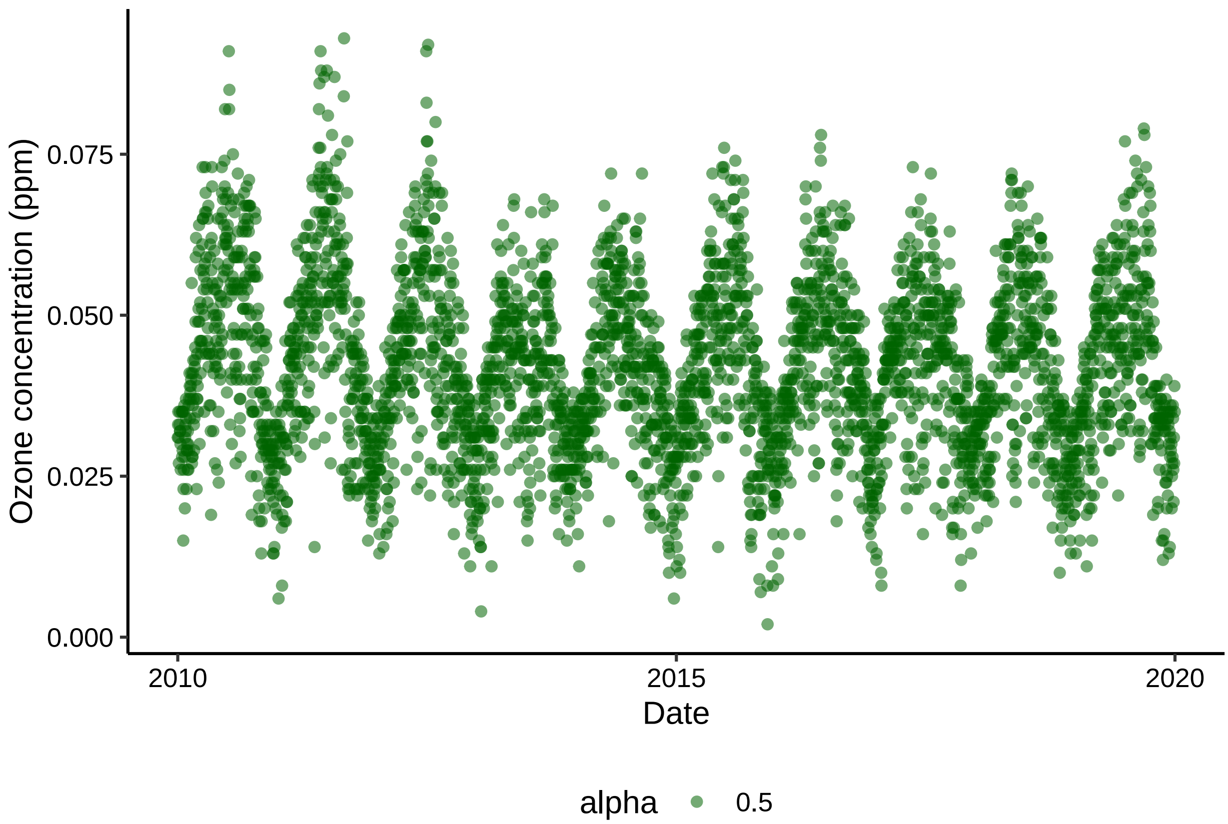
Visualize

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.

```r
#library(RColorBrewer)
```

```r
GaringerOzonePlot <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration, alpha = 0.5)) +
  geom_point(color = "dark green") +
  labs(x = "Date", y = "Ozone concentration (ppm)") +
  scale_color_brewer(palette = "Paired")
print(GaringerOzonePlot)
```

```
## 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 this case, linear interpolation is the most appropriate method due to temporal autocorrelation. Missing ozone concentrations are likely to be close to those on the preceding and follow days, so using linear interpolation is logical as this method fills data gaps with values that fall between the previous and next measurement. The piecewise constant method is inappropriate as missing ozone concentrations are not necessarily equal to measurements made nearest to that date. Additionally, there are not large chunks of missing data in the dataset, so the spine interpolation method also seems inappropriate in this case.

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: The seasonal Mann-Kendall is the most appropriate time series analysis to run in this case because it is the only test that assumes seasonality. Our plot above seems to exhibit a seasonal trend to the data, so the seasonal Mann-Kendall analysis is the most fitting.

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.Con

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

# 10
GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration, frequency = 12,
                      start = c(2010, 01, 01), end = c(2019, 12, 31))

# 11
GaringerOzone.trend <- smk.test(GaringerOzone.monthly.ts)
GaringerOzone.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
```

```
summary(GaringerOzone.trend)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
```

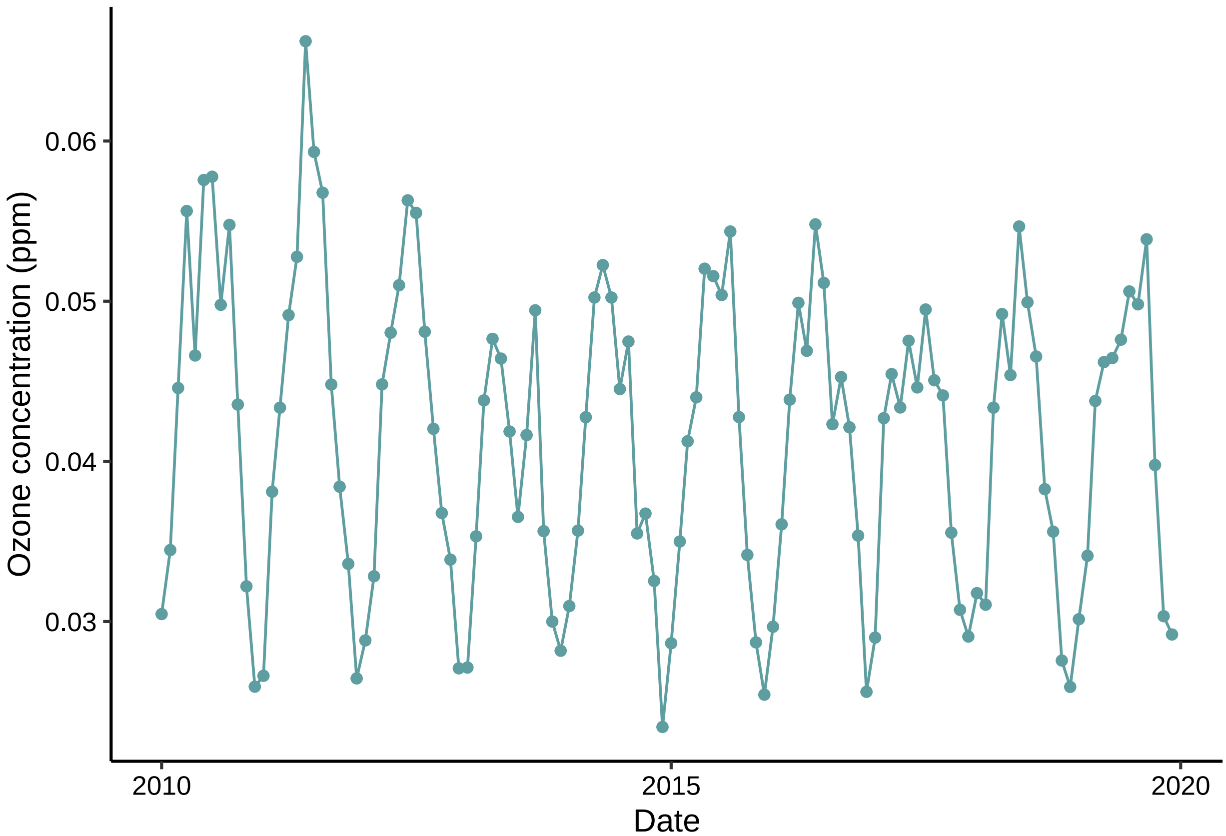
```
##
## H0
##                       S  varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 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
sea.sens.slope(GaringerOzone.monthly.ts)
```

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

```r
# 13

GaringerOzonePlot.monthly <-
  ggplot(GaringerOzone.monthly,
         aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
   geom_point(color = "cadet blue") +
  geom_line(color = "cadet blue") +
  labs(x = "Date", y = "Ozone concentration (ppm)")
print(GaringerOzonePlot.monthly)
```



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: There is a significant monotonic trend in ozone concentration data at Garinger High School over the period of 2010-2019 (seasonal Mann-Kendall: z=-1.963, p=0.04965). The slope of this trend indicates that ozone concentrations are slightly decreasing over the period of 2010-2019 at Garinger High School (Sen's Slope=-0.000207152). We did not detect any seasonal trends across individual months in the dataset.