Assignment 8: Time Series Analysis

Nikki Shintaku

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/nikkishintaku/Desktop/Environmental872/Environmental_Data_Analytics_2020"
```

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

#Set a ggplot theme
mytheme <- theme_stata(base_size = 14, base_family = "sans", scheme = "s2mono") +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)

#import datasets
```

```r
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")
```

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 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
#Using rbind to combine datsets; all datasets have the same column titles
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'))
names(Days)[1] <- "Date"
class(Days$Date)
```

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

```r
# 6
GaringerOzone <- left_join(Days, GaringerOzone_wrangled)
```

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

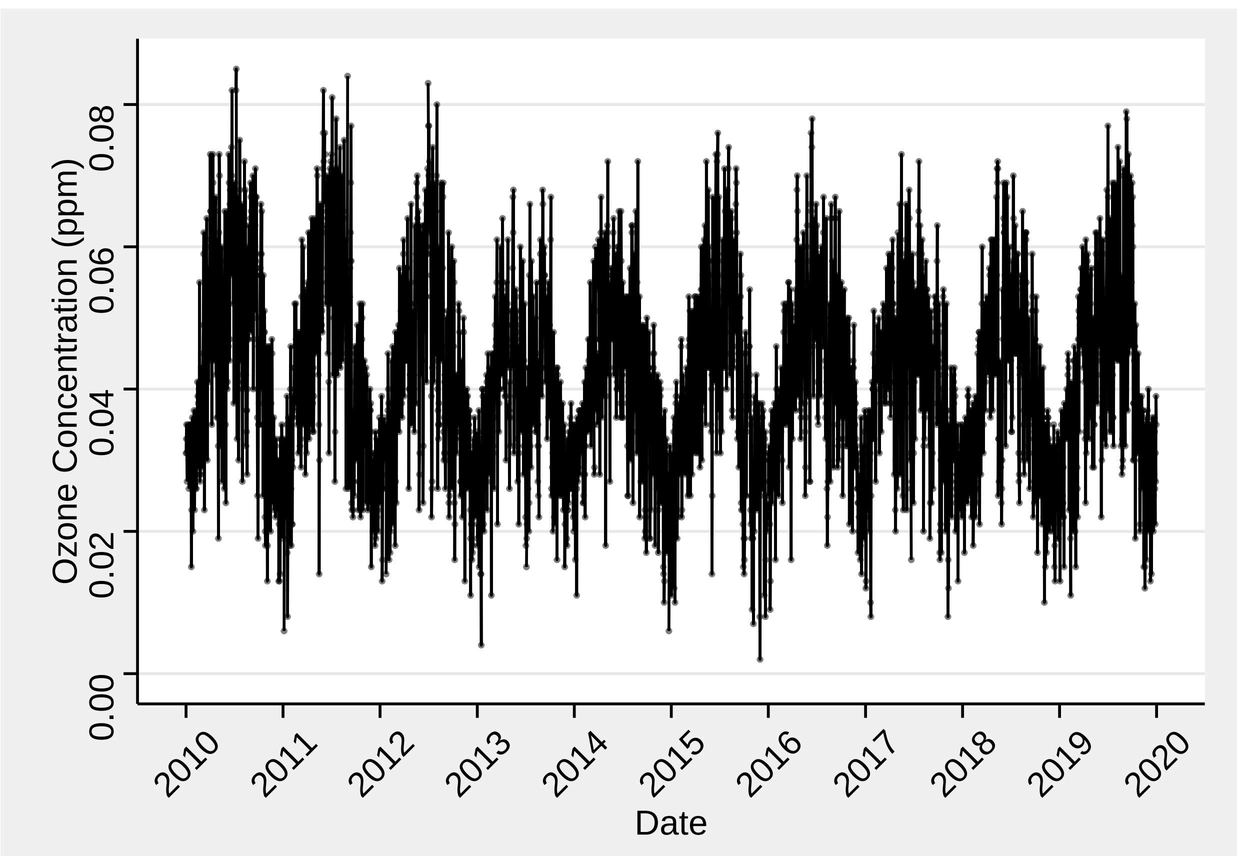
Visualize

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
Ozoneplot <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point(size = 0.5, alpha = 0.5) +
  geom_line() +
  labs(y = "Ozone Concentration (ppm)") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  ylim(0, 0.085) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

plot(Ozoneplot)
```

```
## Warning: Removed 73 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: We shouldn't use a piecewise constant because we don't want to assume that missing data is equal to the measurement made nearest to that date. Ozone concentration can vary from day to day so that might not be the best interpolation method. In addition, spline interpolation is not used because our data looks linear rather than quadratic so spline interpolation may not give us the best results.

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: Seasonal Mann-Kendall is appropriate because we don't want to assume temporal autocorrelation, but we are using monthly data so there is seasonality. We want to see if ozone concentration has changed over the years while incorporating a seasonal component.

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
#linear interpolation using na.approx
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(Ozone.Concentration.mean = 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$Ozone.Concentration.mean, frequency = 12,
                        start = c(2010, 1, 1), end = c(2019, 12, 1))
```

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

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

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

```r
GaringerOzone.monthly.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
```

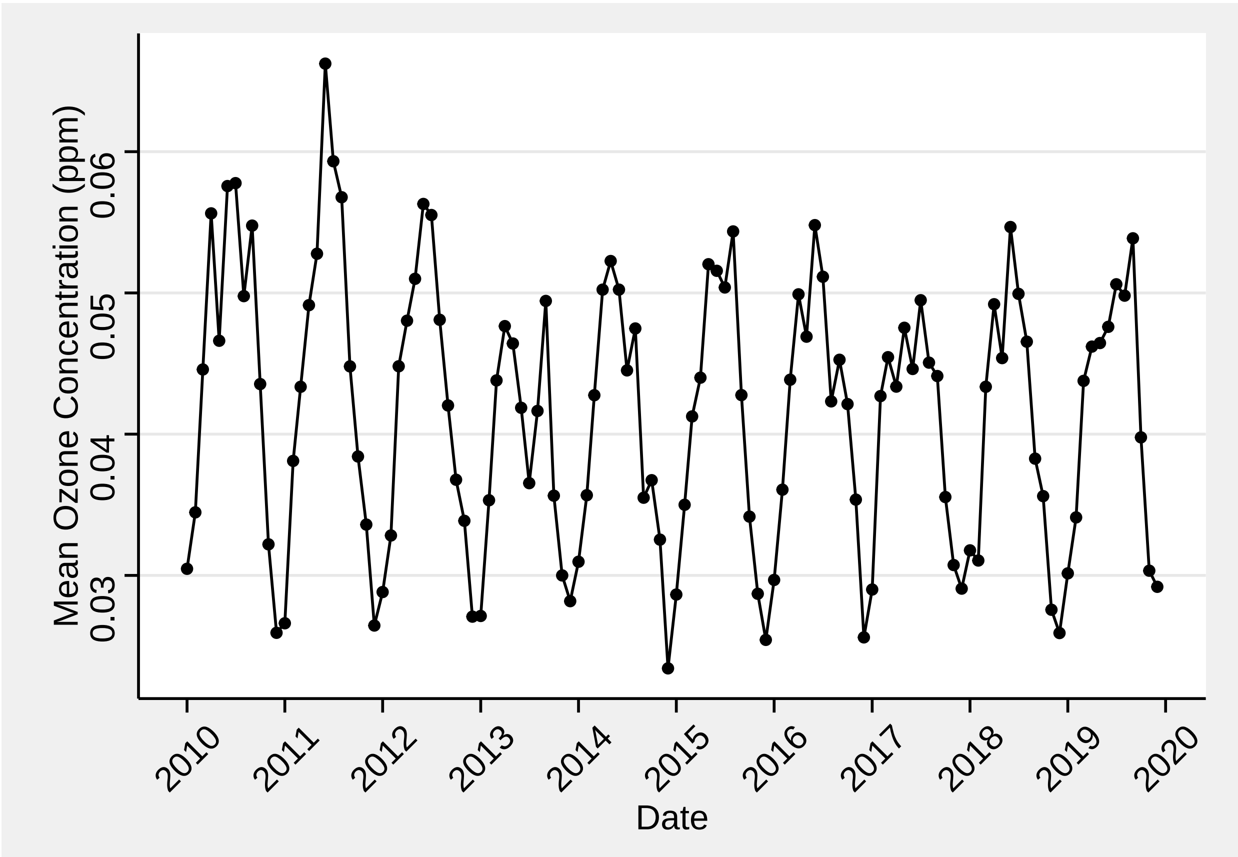
```r
summary(GaringerOzone.monthly.trend)
```

```
##
##  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
# 13
GaringerOzone.monthly.plot <-
  ggplot(GaringerOzone.Monthly, aes(x = Date, y = Ozone.Concentration.mean)) +
  geom_point() +
```

```
  geom_line() +
  labs(y = "Mean Ozone Concentration (ppm)") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))

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: There is a low to no significant trend with ozone concentrations over time at Garinger High School. There is no significant seasons that are an indicator of ozone concentration. (Seasonal Mann-Kendall, z = -1.963, p-value = 0.04965)