Assignment 8: Time Series Analysis

Thomas Hancock

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 - Set up session
getwd() # Check working directory
```

```
## [1] "C:/Users/thoma/Thomas/2018 Grad School/Duke MEM/ENV 872/Environmental_Data_Analytics_2020"
```

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

myTheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top") # Define a theme based off of the classic theme

theme_set(myTheme) # Set defined theme to default

# Load all 10 Ozone datasets
GaringerFiles = list.files(path = "./Data/Raw/Ozone_TimeSeries/", pattern="*.csv", full.names=TRUE)

GaringerOzoneList <- lapply(GaringerFiles, read.csv) # Creates a list of dataframes
```

```
NameList <- list()
for (i in (0:9)) (NameList[i+1] = paste("GaringerOzone201",i, sep = "")) # Create names
names(GaringerOzoneList) <- NameList # Name each dataframe within the list
```

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.

```
# 2 - Combine all data frames into one using rowbind
GaringerOzone = data.frame()
for (i in 1:length(GaringerOzoneList)) {
  GaringerOzone <- rbind(GaringerOzone,GaringerOzoneList[[i]])
}

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

# 4 - Select for only three columns
GaringerOzone <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5 - Create data frame with all days from 1/1/2010 through 12/31/2019
Days <- as.data.frame(seq.Date(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"

# 6 - Join the data frames so there is an entry for each day
GaringerOzone <- left_join(Days, GaringerOzone)
```

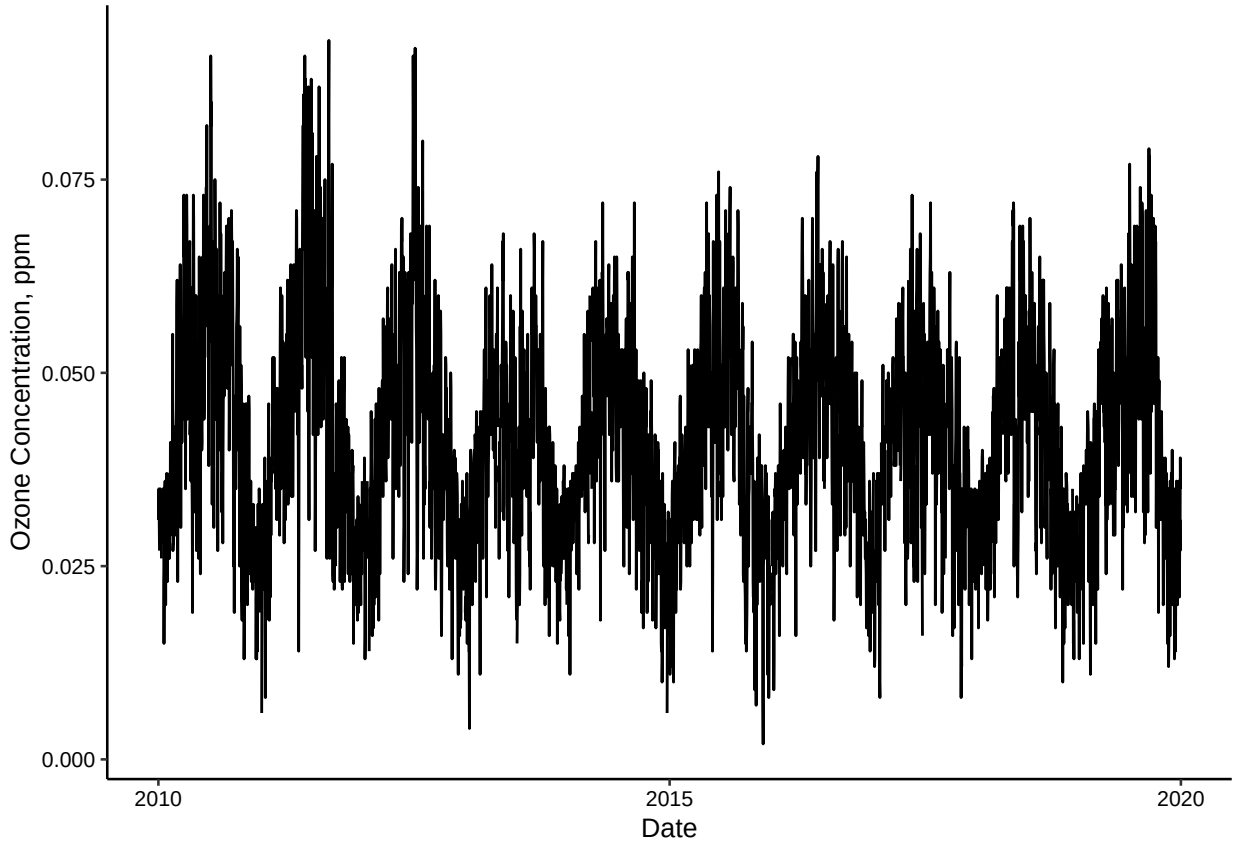
```
## Joining, by = "Date"
```

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.

```
# 7 - Create a line plot for daily ozone concentration
OzonePlot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(x = "Date", y = "Ozone Concentration, ppm")

print(OzonePlot)
```



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: Because ozone concentrations are continuous, it makes sense to not use a piecewise constant interpolation. At one point between the two data points, the ozone concentration would have to pass through the average of the two. (In essence, it is reasonable to assume that the trend between the two endpoints is similar to the trend between the intermediate point and the endpoints.) The spline interpolation is unnecessary because the data points are close enough together/there are enough of them that a linear interpolation probably approximates the shape of the ozone concentration well enough. Avoiding spline interpolations also avoids any concerns over negative values (which are impossible).

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 would expect ozone to be seasonal (usually higher in the summer), so it is best to use an analysis that includes seasonality. The biggest concern with this is that the seasonal Mann-Kendall analysis assumes there is no temporal autocorrelation, which is likely not true. To

help mitigate this autocorrelation (among other reasons), we will aggregate the data into monthly averages.

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 - Fill in missing days using linear interpolation
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)

# 9 - Aggregate monthly data
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = year(Date),
         Month = month(Date)) %>% # Add Year and Month columns
  group_by(Year, Month) %>%
  summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration)) # Find monthly averages

GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$Year,
                                            GaringerOzone.monthly$Month, 1, sep = "-"),
                                      format = "%Y-%m-%d") # Create Date column for 1st of each month


# 10 - Create a time series of the monthly means
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone, frequency = 12,
                          start = c(2010, 1, 1), end = c(2019, 12, 1))


# 11 - Run seasonal Mann-Kendall test
GaringerOzone.trend <- smk.test(GaringerOzone.monthly.ts) # Run test
GaringerOzone.trend # Show test results
```

```
##
##  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) # Show seasonal results
```

```
##
##  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|)
```

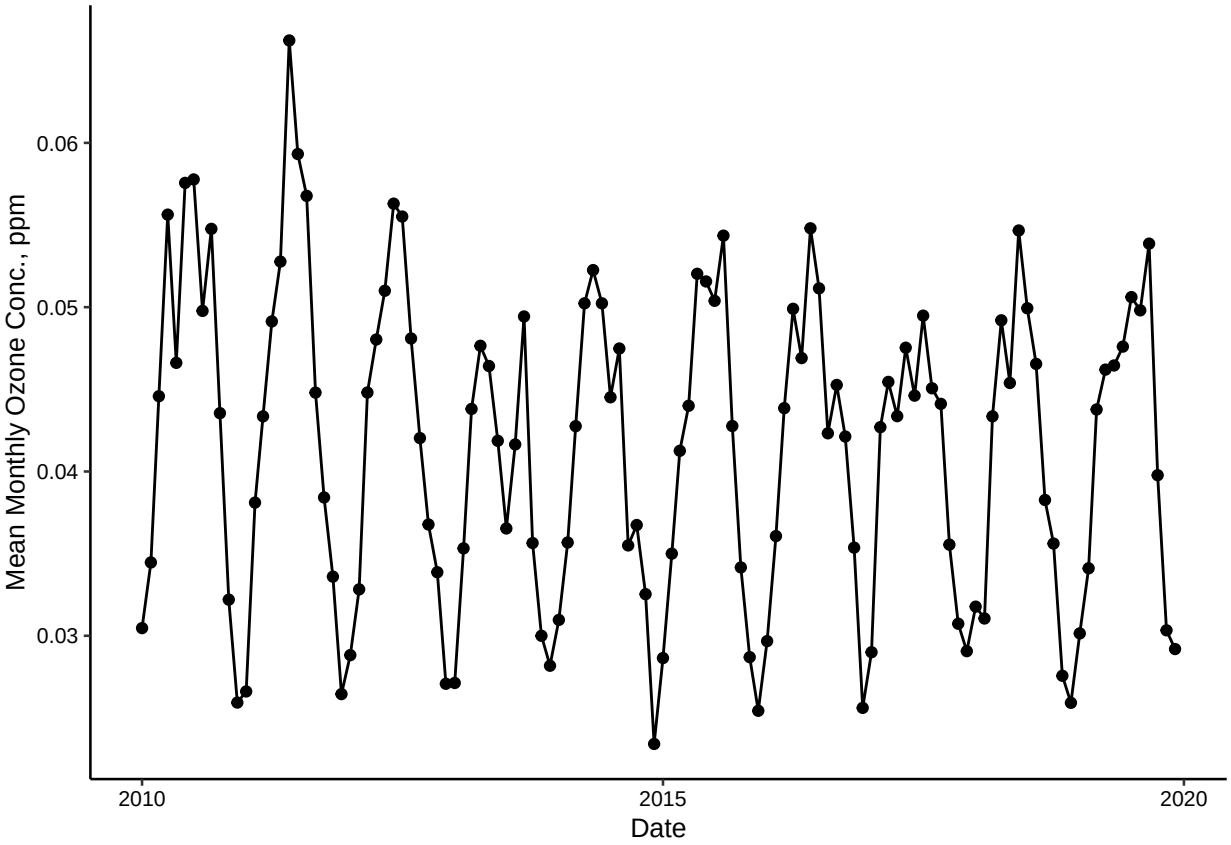
```
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  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
```

```
# 12 - Run Sen's slope analysis
GaringerOzone.slope <- sea.sens.slope(GaringerOzone.monthly.ts)
GaringerOzone.slope
```

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

```
# 13 - Plot monthly ozone data
OzoneMonthPlot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = MeanOzone)) +
  geom_point() + # Show points for monthly averages
  geom_line() + # Connect monthly averages with straight lines
  ylab("Mean Monthly Ozone Conc., ppm")

print(OzoneMonthPlot)
```



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: According to our time series analysis, ozone concentrations at the Garinger High School have changed in a statistically significant way over the 2010s (Seasonal Mann-Kendall, z = -1.963, p = 0.04965). The ozone concentration has decreased over time, with a change of -0.0002 ppm per year. The SMK test did not reveal a significant change for any particular month over time, even though the overall trend was downward.