Assignment 7: Time Series Analysis

Michael Gaffney

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., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#set working directory
setwd("/Users/michaelgaffney/Documents/Duke University/Nicholas School of the Environment/05 Spring 202

#load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(readr)

#set ggplot theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

2. 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). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#2 Batch import files
GaringerOzone <- list.files("/Users/michaelgaffney/Documents/Duke University/Nicholas School of the Envi
  lapply(read_csv) %>%
  bind_rows
```

Wrangle

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.

```
# 3 change dates
GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4 wrangle data and remove unneeded columns
GaringerOzoneSelected <- GaringerOzone %>%
  select(Date, 'Daily Max 8-hour Ozone Concentration', DAILY_AQI_VALUE)
```

```
# 5
#generate sequence of dates in a new dataframe.
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day"))
#rename column with dplyr
Days <- Days %>%
  rename(Date = `seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day")`)
# 6
GaringerOzoneFinal <- left_join(Days, GaringerOzoneSelected)
```

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

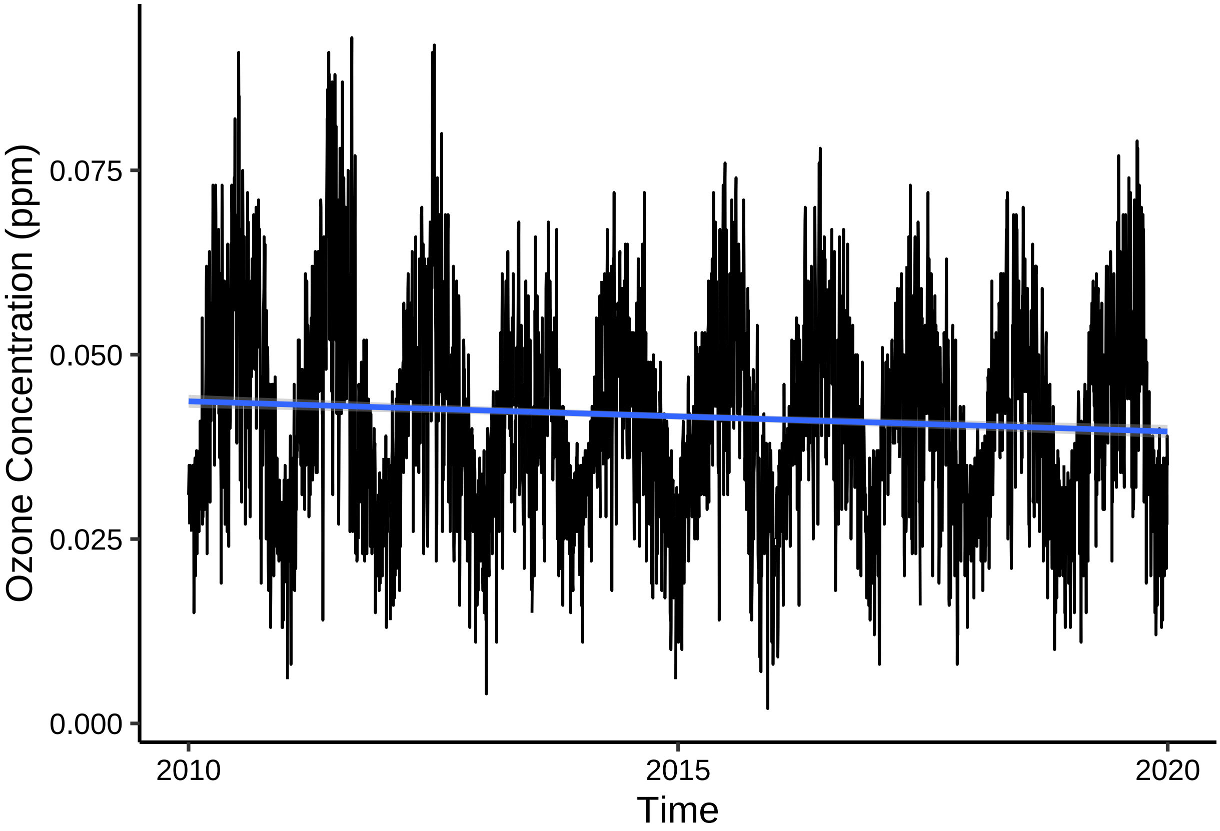
Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
ggplot(GaringerOzoneFinal, aes(x=Date, y=`Daily Max 8-hour Ozone Concentration`)) +
  geom_line() +
  geom_smooth(method = lm) +
  xlab("Time") + ylab("Ozone Concentration (ppm)")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```



Answer: The plot would seem to suggest that there is a negative trend in the data: daily max ozone concentrations appear to be delcining over time.

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?

````
#8
GaringerOzoneInterpolated <- GaringerOzoneFinal %>%
  mutate(`Daily Max 8-hour Ozone Concentration` = zoo::na.approx(`Daily Max 8-hour Ozone Concentration`
````

> Answer: Linear interpolation fills in the gaps between points by fitting a straight line between the nearest points; a spline interpolation would use a quadratic equation to fill in the gaps between the points. Piecewise constant, meanwhile, would fill in the missing data by assuming it is equal to the nearest value. In this case, we used linear interpolation because values between dates are rarely the same, and they are continuous. Because of this, linear would likely produce more accurate 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)

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

````
## `summarise()` has grouped output by 'Year', 'Month'. You can override using the `.groups` argument.
````

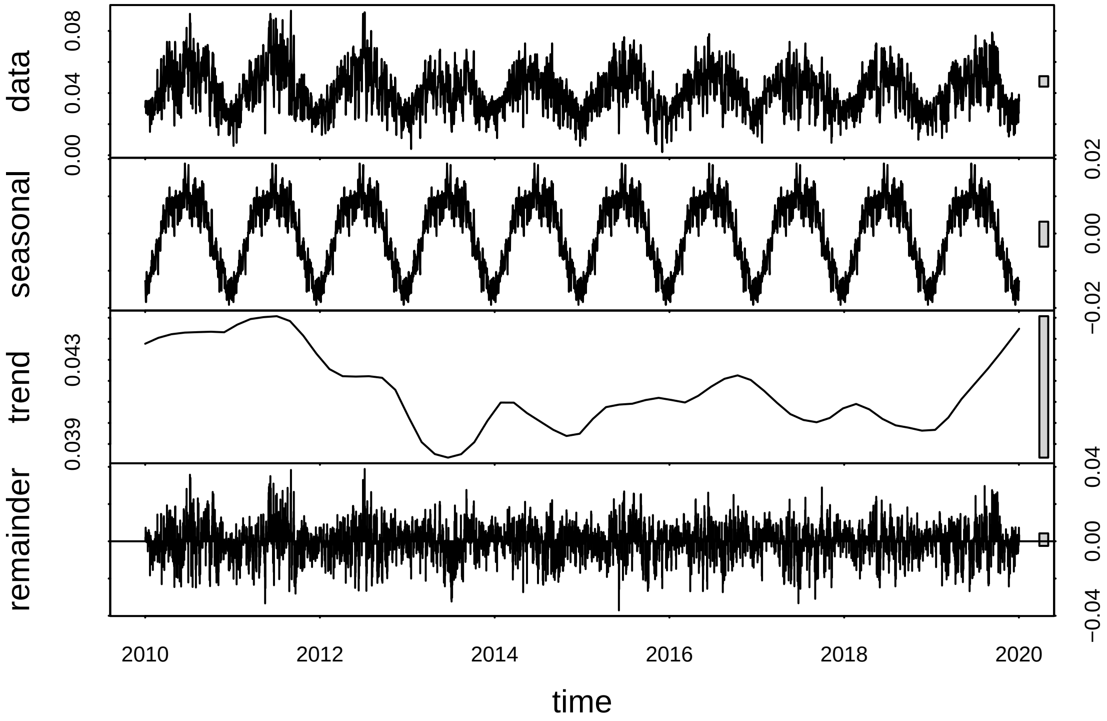
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

````
#10
#daily time series object
first_year <- year(first(GaringerOzoneInterpolated$Date))
first_month <- month(first(GaringerOzoneInterpolated$Date))
first_day <- day(first(GaringerOzoneInterpolated$Date))

GaringerOzone.daily.ts <- ts(GaringerOzoneInterpolated$`Daily Max 8-hour Ozone Concentration`, start = c

#monthly times series object
first_year <- year(first(GaringerOzone.monthly$Month_Year))
first_month <- month(first(GaringerOzone.monthly$Month_Year))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone, start = c(first_year, first_month), fre
````

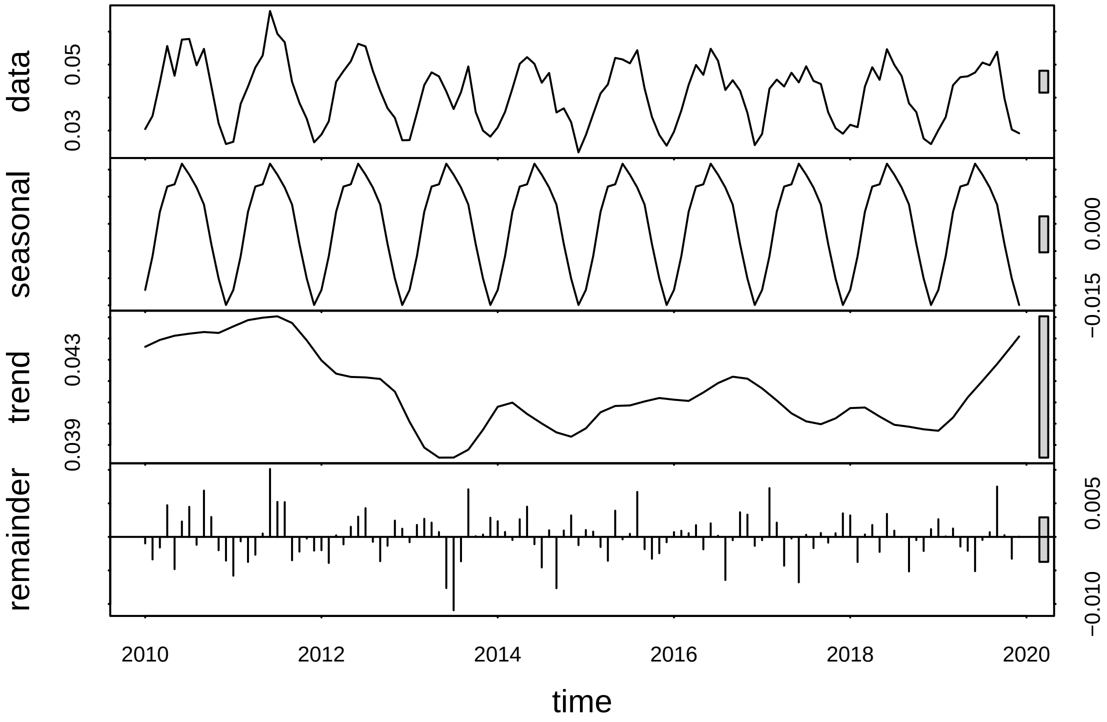
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

````
#11
#decompose daily and monthly
Garinger.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
Garinger.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
````

```
#plot daily and monthly
plot(Garinger.daily.decomposed)
```



```
plot(Garinger.monthly.decomposed)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```r
#12
#run Mann Kendall test on the original time series object
GaringerOzone_Kendall <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
#get results
summary(GaringerOzone_Kendall)
```

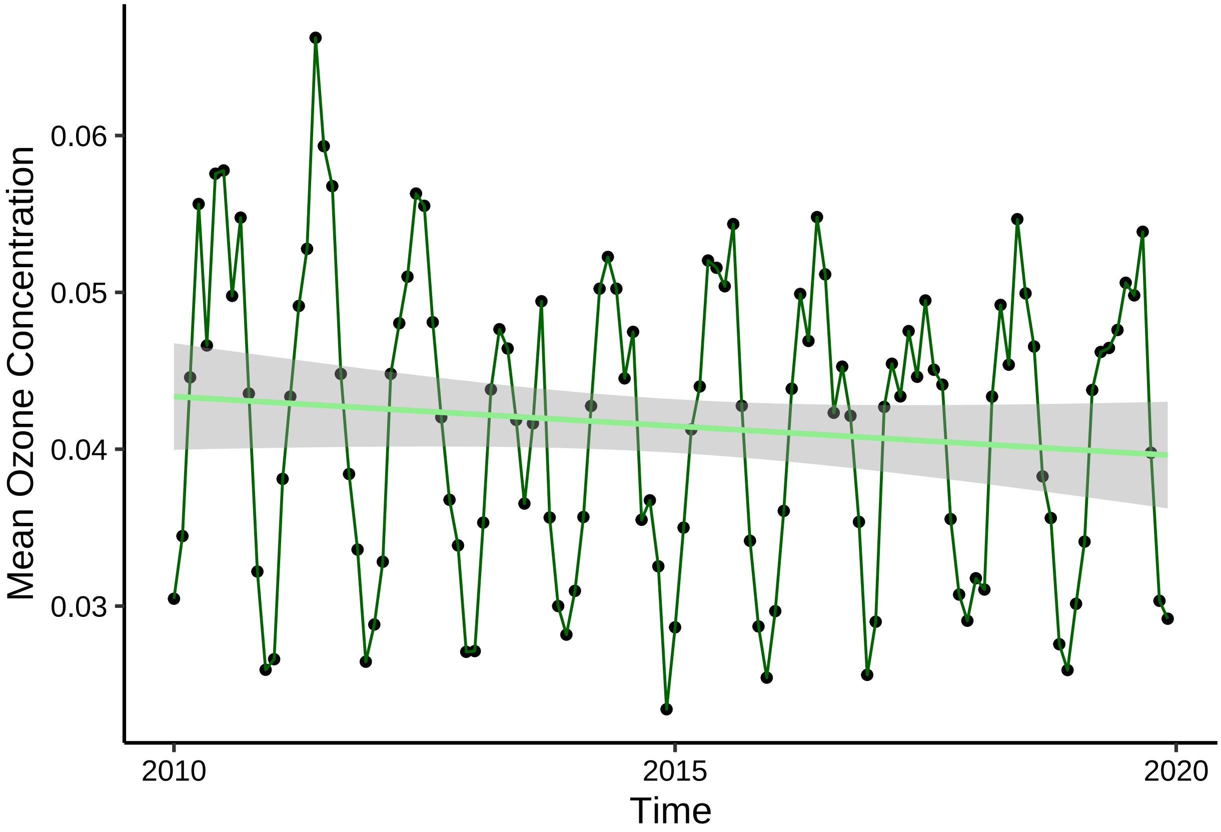
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The Mann Kendall test is appropriate for this dataset because it is nonparametric, and it is likely that the distribution of the ozone data is non-normal. We specifically want the seasonal Mann Kendall test becuase there is a seasonal trend in the ozone data (ie, ozone concetrations rise and fall as the seasons change, which we can see from the decomposition plot).

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```r
# 13
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = MeanOzone)) +
  geom_point() +
  geom_line(color = "darkgreen") +
  geom_smooth(method=lm, color = "lightgreen") +
  xlab("Time") + ylab("Mean Ozone Concentration")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



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: My research question was whether Ozone concentrations have changed at this station over the course of the 2010s. Analysis of the data suggests that there has been a significant decline in mean ozone concetration. A Mann Kendall test yielded a tau of -0.143 and a p-value of 0.047, suggesting that there is a signficant decline (although only marginally so, given how close the value is to .05).

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.
16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
GaringerOzone.monthly.components <- as.data.frame(Garinger.monthly.decomposed$time.series[,1:3])
GaringerOzone.monthly.subtracted <- GaringerOzone.monthly.ts - GaringerOzone.monthly.components$seasona

#16
GaringerOzone_Kendall_2 <- Kendall::MannKendall(GaringerOzone.monthly.subtracted)
#get results
summary(GaringerOzone_Kendall_2)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: Running a standard Mann Kendall with the seasonal component already substracted from the time series, both the tau and p-value change. The tau returns -0.165 and the pvalue is 0.0075. The negative relationship between time and ozone concentrations proves to be even more statistically significant. In other words, we once again reject the null hypothesis that there is no change over time.