Assignment 7: Time Series Analysis

Jack Carpenter

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

```
#1
getwd()
```

```
## [1] "/Users/Jack/Documents/Duke/Spring 2022/Environmental Data Analytics/Environmental_Data_Analytics
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     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
```

```r
library(zoo)
```

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

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

```r
library(trend)

theme_set(theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right"))
```

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.

```r
#2
GaringerO3_2010 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2011 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2012 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2013 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2014 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2015 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2016 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2017 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2018 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
           stringsAsFactors = TRUE)
GaringerO3_2019 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
           stringsAsFactors = TRUE)

GaringerOzone <- rbind(GaringerO3_2010, GaringerO3_2011, GaringerO3_2012,
                       GaringerO3_2013, GaringerO3_2014, GaringerO3_2015,
                       GaringerO3_2016, GaringerO3_2017, GaringerO3_2018,
```

```
                     GaringerO3_2019)
dim(GaringerOzone)
```

```
## [1] 3589   20
```

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

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

```
# 4 Data Cowboy time
GaringerO3 <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
dim(GaringerO3)
```

```
## [1] 3589    3
```

```
# 5 Daily Dataframe
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = 1))
names(Days)[1] <- "Date"
dim(Days)
```

```
## [1] 3652    1
```

```
# 6 Left Shark
GaringerOzone <- left_join(Days, GaringerO3)
```

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

```
#have to have Days listed first for some reason
dim(GaringerOzone)
```

```
## [1] 3652    3
```

Visualize

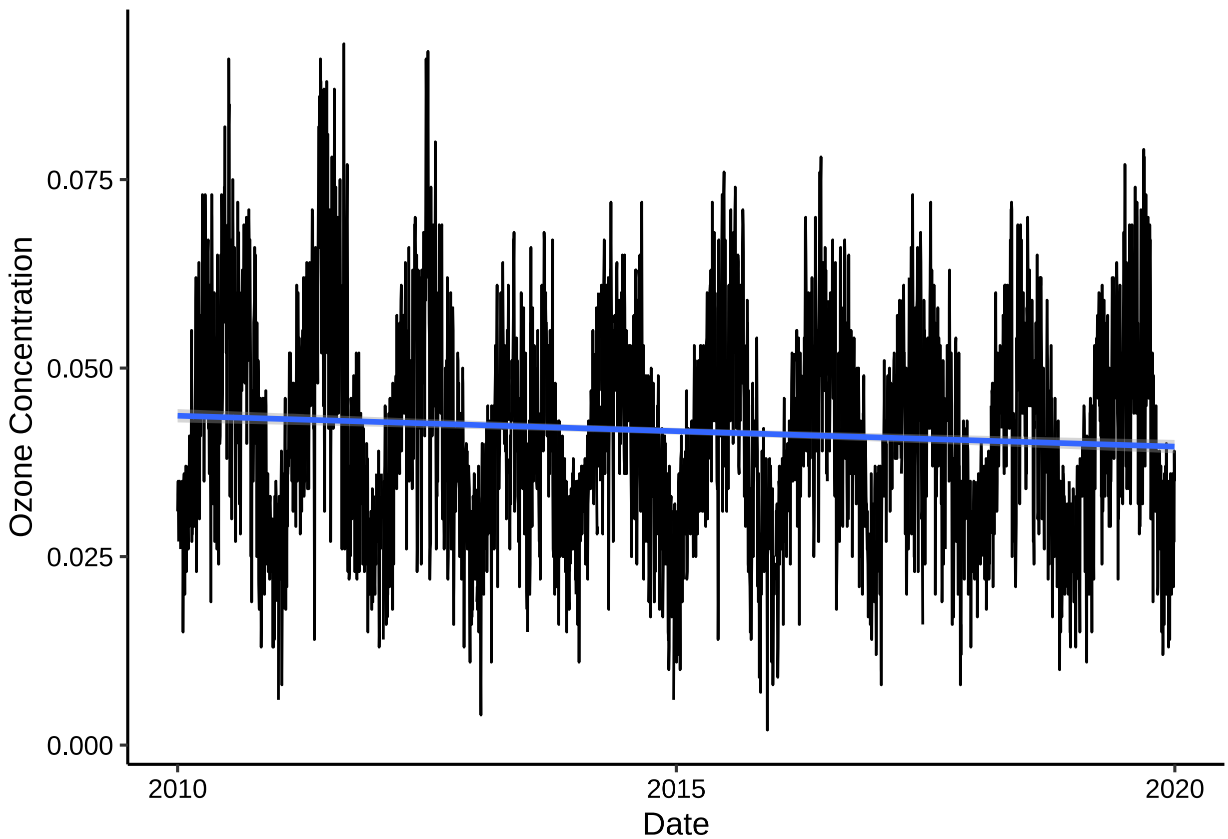
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7 Plot it out now y'all
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
```

```
  geom_smooth(method = lm) +
  labs(x = "Date", y = "Ozone Concentration")
```

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

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



Answer: Yes, the plot suggests a decreasing trend in concentration 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 Linear Interpolation
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: We didn't use a piecewise because the "nearest neighbor" would throw off the data by making an assumption of closer value. The spline would also potentially be over complicated between two points, because Ozone levels probably do not drop exponentially between days, given the data.

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 New DF, who dis?
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Month = month(Date),
         Year = year(Date)) %>%
  mutate(Month_Year = my(paste(Month, "-", Year))) %>%
  group_by(Month_Year) %>%
  summarize(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone.monthly
```

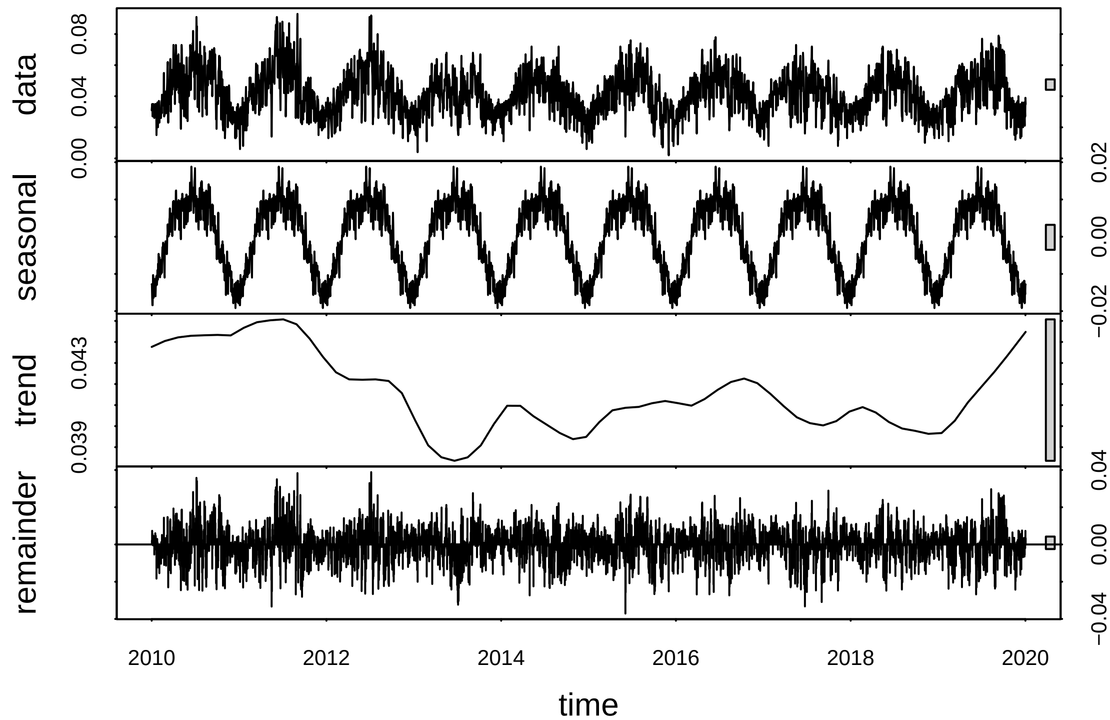
```
## # A tibble: 120 x 2
##    Month_Year mean_Ozone
##    <date>          <dbl>
##  1 2010-01-01     0.0305
##  2 2010-02-01     0.0345
##  3 2010-03-01     0.0446
##  4 2010-04-01     0.0556
##  5 2010-05-01     0.0466
##  6 2010-06-01     0.0576
##  7 2010-07-01     0.0578
##  8 2010-08-01     0.0498
##  9 2010-09-01     0.0548
## 10 2010-10-01     0.0435
## # ... with 110 more rows
```

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.

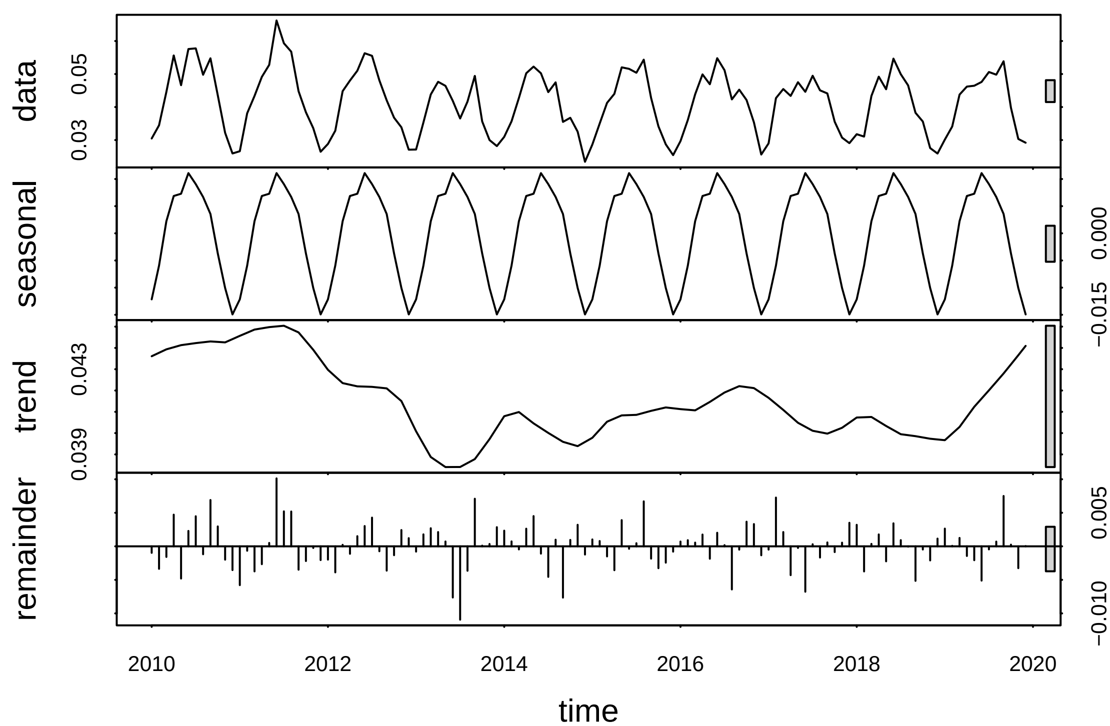
```
#10 Time Travel
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,01),
                             frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_Ozone,
                               start = c(2010,01),
                               frequency = 12)
```

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

```
#11 Break it down now y'all
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts,
                                    s.window = "periodic")
plot(GaringerOzone.monthly.decomp)
```



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

```
#12 Mann-Kendall (another Jenner?)
GaringerOzone.monthly.trend <-
```

```
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

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

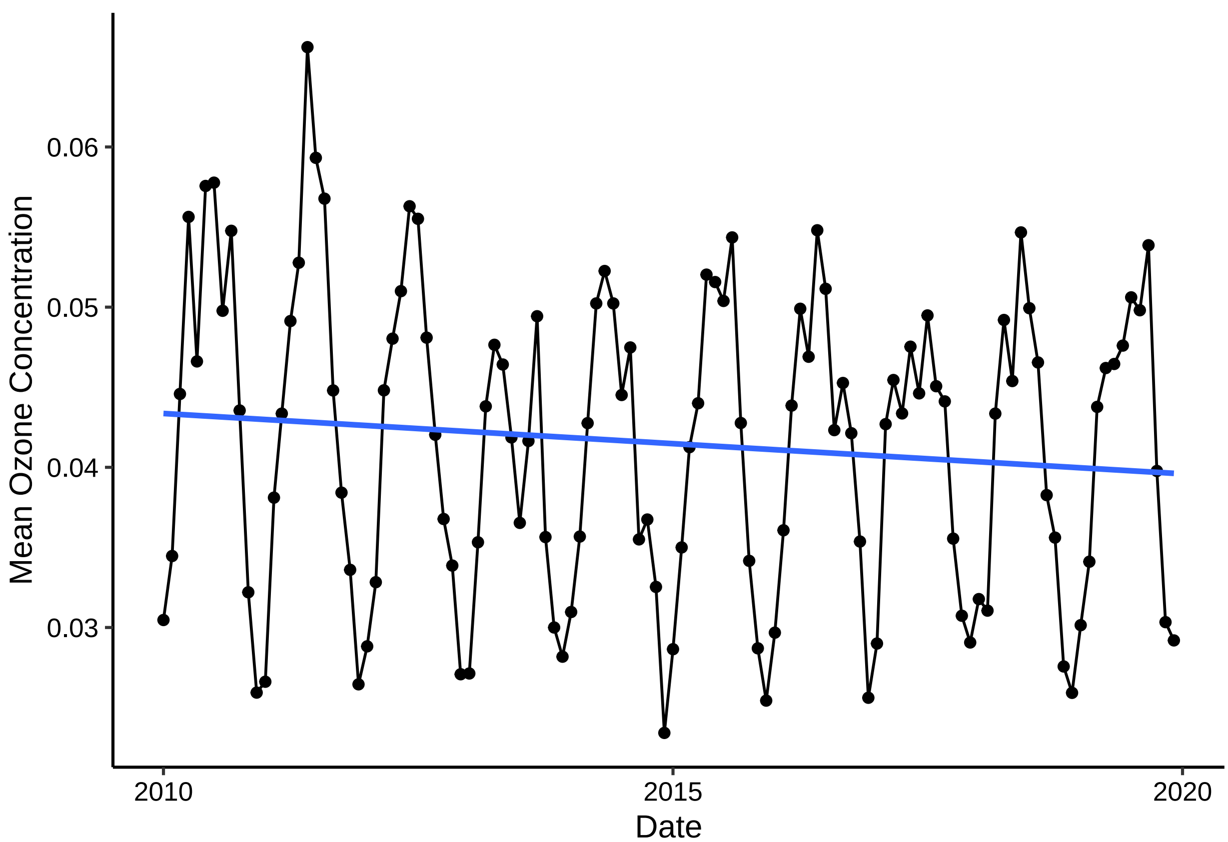
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Seasonal mann-kendall is the only one that evaluates seasonality, and our data is very seasonal.

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.

```
# 13 Plots! Plots! Plots! Everybody!
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = mean_Ozone))+
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Ozone Concentration") +
  geom_smooth(method = lm, se = FALSE)
```

```
## `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: We reject the null hypothesis that the data is stationary, there is a decreasing trend, but just barely (p-value = 0.0467).

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 Get rid of the seasons (aka Florida-ization)
#GaringerOzone.monthly.decomp
GaringerOzone.monthly.components <-
  as.data.frame(GaringerOzone.monthly.decomp$time.series[,2:3])

#16 Mann-Kendall Season-Free (A Floridian Jenner?)
GaringerOzone.monthly.no_seasons.ts <-
  ts(GaringerOzone.monthly.components$trend,
     start = c(2010,1),
     frequency = 12)
GaringerOzone.monthly.no_seasons.trend <-
  Kendall::MannKendall(GaringerOzone.monthly.no_seasons.ts)
summary(GaringerOzone.monthly.no_seasons.trend)
```

```
## Score =  -1922 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

Answer: The season-free Mann-Kendall has a much smaller p-value (1.316e-05) than the p-value for the seasonal Mann-Kendall (0.0467) and so shows a much more significant trend.