Assignment 7: Time Series Analysis

GuruBandaa Khalsa

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 Tuesday, November 4 at 5:00 pm.

Set up

1. Set up your session:

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

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.

```
#1. I checked my working directory using the getwd() function, loaded the
#   `tidyverse,' `lubridate,' 'zoo,' and 'trend' packages, and set knitting
#   settings.
getwd()
```

```
## [1] "/Users/survivormangb/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analy
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2
```

```
library(lubridate)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.1.2
```

```
library(trend)
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=80), tidy=TRUE)

# I assigned a pre-built theme from Ggplot as a common theme across all plots
# in my analysis session.
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

#2. I imported the ten datasets from the Ozone_TimeSeries folder in the Raw
#   data folder and then combined them into a single dataframe
OTS1 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS2 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS3 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS4 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS5 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS6 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS7 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS8 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS9 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
OTS10 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA

GaringerOzone <- rbind(OTS1, OTS2, OTS3, OTS4, OTS5, OTS6, OTS7, OTS8, OTS9, OTS10)
```

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. I set my date column as a date class.
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4. I wrangled my dataset so that it only contains the columns Date,
# Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE
GaringerOzone.wrangle <- GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5. I created a new data frame that contains a sequence of dates from
# 2010-01-01 to 2019-12-31 named 'Days' and renamed the column name in Days to
# 'Date.'
Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"),
    by = 1))
colnames(Days) = "Date"

# 6. I used a `left_join` to combine the data frame
GaringerOzone <- left_join(Days, GaringerOzone.wrangle, by = c("Date"))
```

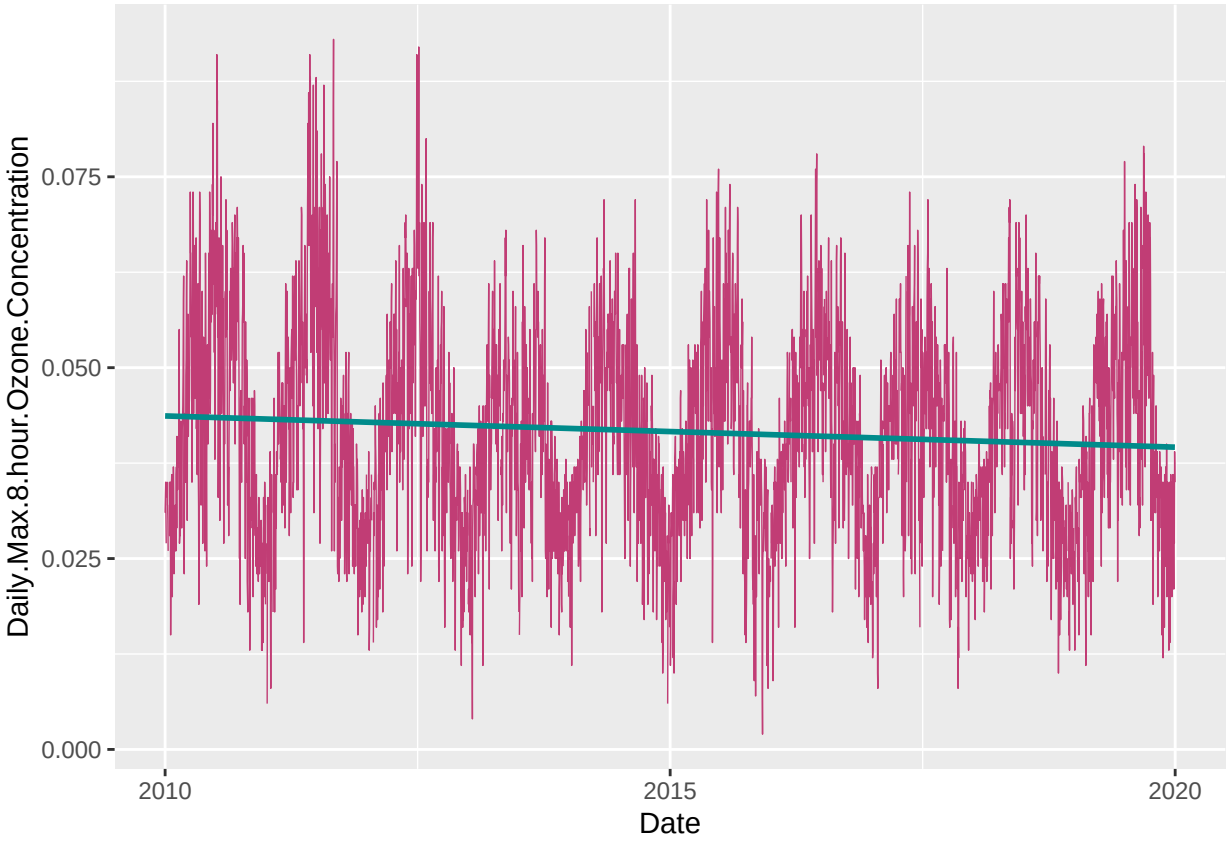
Visualize

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. I created a line plot depicting ozone concentrations over time with actual
# concentrations in ppm, not AQI values.
ozonePlot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line(aes(y = Daily.Max.8.hour.Ozone.Concentration, x = Date), size = 0.25,
        color = "#c13d75ff") + geom_smooth(method = "lm", se = FALSE, col = "cyan4")
print(ozonePlot)
```

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

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



Answer: My plot suggests a slight negative trend in ozone 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. I used a linear interpolation to fill in missing daily data for ozone
# concentration.
GaringerOzone.clean <- GaringerOzone %>%
    mutate(Ozone.Conc = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We did not use a piecewise constant because that assumes missing data to the measurement made nearest to that date. We did not use a spline interpolation because that uses a quadratic function to interpolate rather than drawing a straight line, like the linear interpolation. In the linear approach, a straight line is drawn between the known points and missing data is assumed to fall between the previous and next measurement. This method suits our 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)

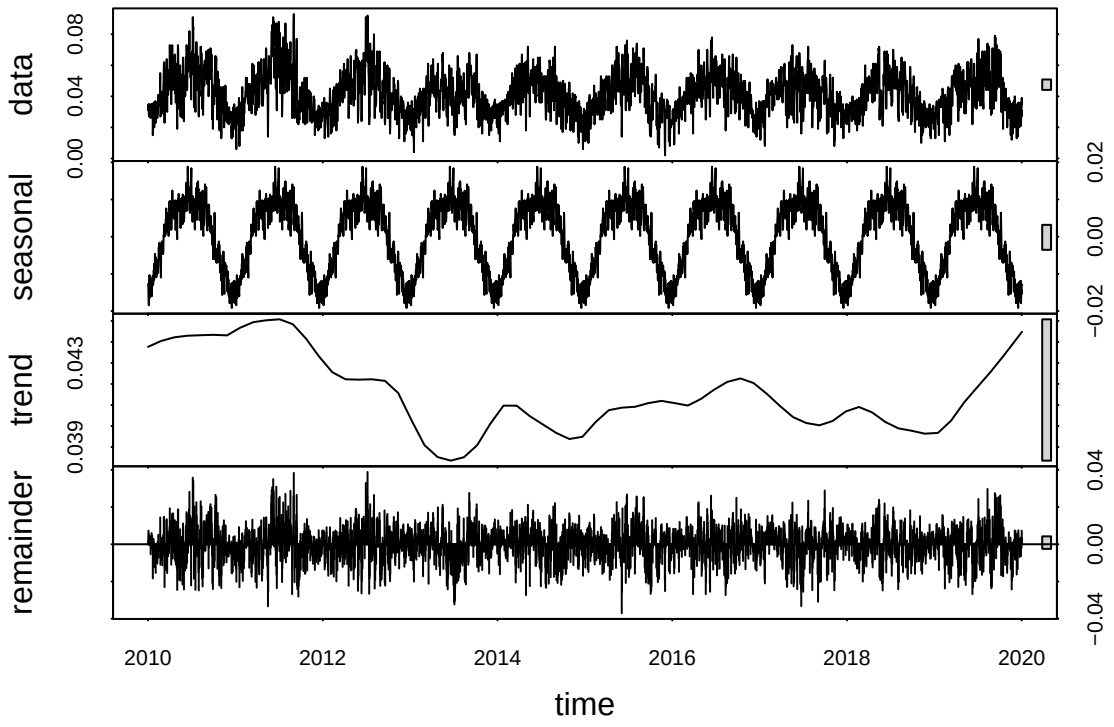
```
# 9. I created a new data frame called `GaringerOzone.monthly` that contains
# aggregated data: mean ozone concentrations for each month.
GaringerOzone.monthly <- GaringerOzone.clean %>%
    mutate(month = month(Date), year = year(Date)) %>%
    mutate(Month_year = my(paste0(month, "-", year))) %>%
    select(Month_year, Ozone.Conc) %>%
    group_by(Month_year) %>%
    summarise(Mean_PPM = mean(Ozone.Conc), na.rm = TRUE)
```

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.

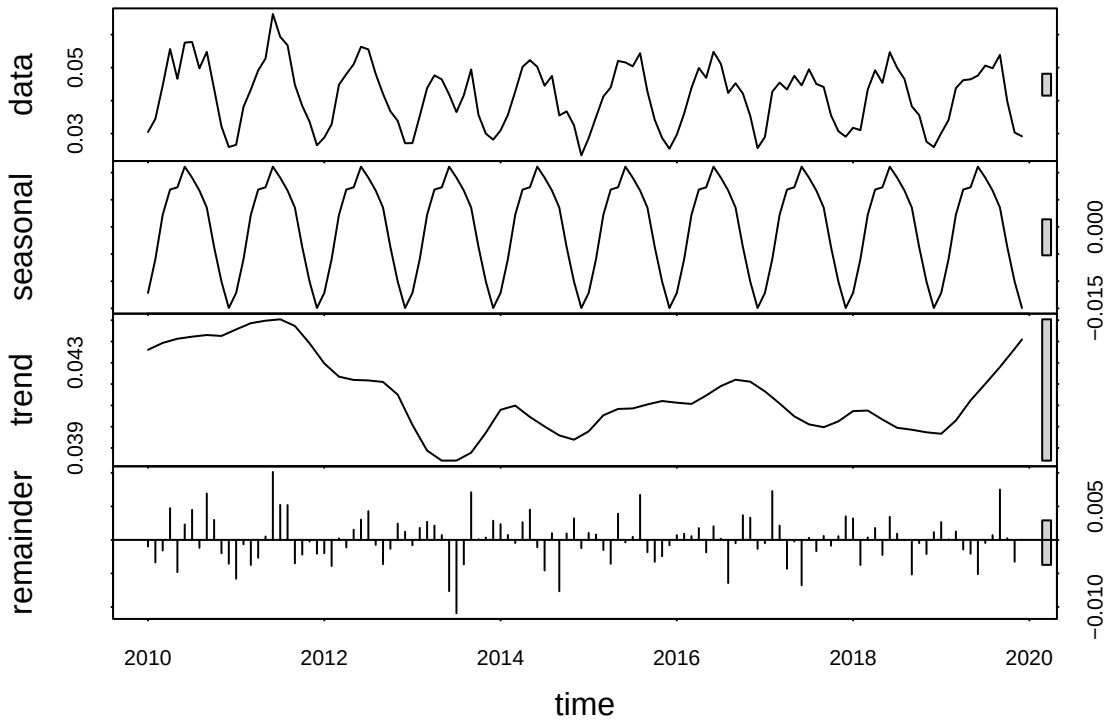
```
# 10. I generated two time series objects, one based on the dataframe of daily
# observations and another on the monthly average ozone values.
f_month1 <- month(first(GaringerOzone.clean$Date))

f_year1 <- year(first(GaringerOzone.clean$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone.clean$Ozone.Conc, frequency = 365, start = c(f_year1,
    f_month1))


f_month2 <- month(first(GaringerOzone.monthly$Month_year))

f_year2 <- year(first(GaringerOzone.monthly$Month_year))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_PPM, frequency = 12, start = c(f_year2,
    f_month2))
```

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

```
# 11. I decomposed the daily and the monthly time series objects and plot the
# components using the `plot()` function.
GaringerOzone.daily.Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.Decomposed)
```




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

plot(GaringerOzone.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?

```
# 12. I ran a monotonic trend analysis for the monthly Ozone series.
GaringerOzone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.trend)
```

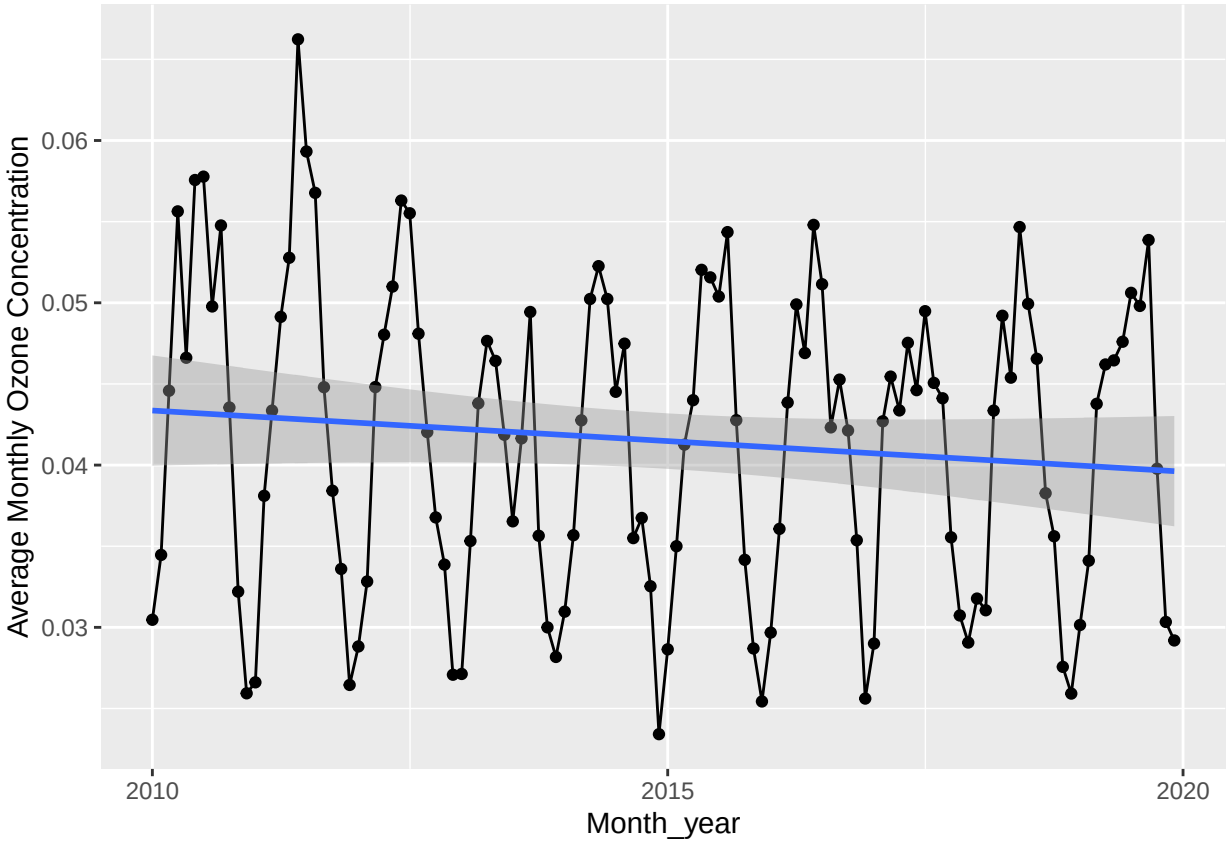
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall is the most appropriate because it accounts for seasonality. The traditional Mann-Kendall does not separate out seasonal trends.

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. I created a plot depicting mean monthly ozone concentrations over time,
# with both a geom_point and a geom_line layer.
GaringerOzone.plot <- ggplot(GaringerOzone.monthly, aes(x = Month_year, y = Mean_PPM)) +
    geom_point() + geom_line() + ylab("Average Monthly Ozone Concentration") + geom_smooth(method = lm)
print(GaringerOzone.plot)
```

```
## `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: Have ozone concentrations changed over the 2010s at this station? Ozone concentrations have changed over the 2010s at this station. They have gradually decreased from approximately 0.044 to 0.042. Since the S value from the seasonal Mann-Kendall test is -77, which is less than 0, this indicates that there is a downward trend. The tau value also indicates a negative trend, since its value is approximately -0.14 (it is negative within the range -1 to 1). Lastly, the two-sided p-value is 0.046724, which is significant. Therefore, the relationship between ozone concentration and time is significant.

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. I subtracted the seasonal component from the `GaringerOzone.monthly.ts.'
GaringerOzone.monthly.nonseas <- GaringerOzone.monthly.ts - GaringerOzone.monthly.Decomposed$time.series
    1]

# 16. I ran the Mann-Kendall test on the non-seasonal Ozone monthly series.
GaringerOzone.nonseas.trend <- Kendall::MannKendall(GaringerOzone.monthly.nonseas)
summary(GaringerOzone.nonseas.trend)
```

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

Answer: When comparing the results for the Mann-Kendall test on the non-seasonal Ozone monthly series versus the Seasonal Mann-Kendall test, the scores are -1179 and -77, Var(Scores) of 194365.7 and 1499, denominators of 7139.5 and 539.4972, tau values of -0.165 and -0.143, and 2-sided p-values of 0.0075402 and 539.4972. Both tests produce results that indicate the same negative trend. The results that stand out most are the negative scores and low p-values.