Assignment 7: Time Series Analysis

Eva May

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, March 16 at 11:59 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
setwd("/Users/evamay/Desktop/ENV872/Environmental.Data.Analytics.2021")
getwd()
```

```
## [1] "/Users/evamay/Desktop/ENV872/Environmental.Data.Analytics.2021"
```

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(zoo)
library(trend)
library(tidyr)

EMtheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "dark grey"),
        legend.position = "top")
theme_set(EMtheme)

#2
```

```
Garinger10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", header=TRUE)
Garinger11 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", header=TRUE)
Garinger12 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", header=TRUE)
Garinger13 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", header=TRUE)
Garinger14 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", header=TRUE)
Garinger15 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", header=TRUE)
Garinger16 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", header=TRUE)
Garinger17 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", header=TRUE)
Garinger18 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", header=TRUE)
Garinger19 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", header=TRUE)

Garinger10$Site.ID <- as.numeric(Garinger10$Site.ID)
Garinger11$Site.ID <- as.numeric(Garinger11$Site.ID)
Garinger12$Site.ID <- as.numeric(Garinger12$Site.ID)
Garinger13$Site.ID <- as.numeric(Garinger13$Site.ID)
Garinger14$Site.ID <- as.numeric(Garinger14$Site.ID)
Garinger15$Site.ID <- as.numeric(Garinger15$Site.ID)
Garinger16$Site.ID <- as.numeric(Garinger16$Site.ID)
Garinger17$Site.ID <- as.numeric(Garinger17$Site.ID)
Garinger18$Site.ID <- as.numeric(Garinger18$Site.ID)
Garinger19$Site.ID <- as.numeric(Garinger19$Site.ID)
class(Garinger19$Site.ID)
```

```
## [1] "numeric"
```

```
GaringerOzone <- rbind(Garinger10, Garinger11, Garinger12, Garinger13, Garinger14,
                       Garinger15, Garinger16, Garinger17, Garinger18, Garinger19)

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

# 4
GaringerOzone.2 <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
summary(GaringerOzone.2)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
```

```
##  Min.   :2010-01-01   Min.   :0.00200                    Min.   :  2.00
##  1st Qu.:2012-07-03   1st Qu.:0.03200                    1st Qu.: 30.00
##  Median :2015-01-04   Median :0.04100                    Median : 38.00
##  Mean   :2015-01-01   Mean   :0.04163                    Mean   : 41.57
##  3rd Qu.:2017-07-02   3rd Qu.:0.05100                    3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                    Max.   :169.00
```

```
# 5

Days <- as.data.frame(seq(as.Date("2010/01/01"), as.Date("2019/12/31"), by ="day"))
names(Days)[1] <- "Date"

# 6

#x=rows, y = rows and cols
GaringerOzone <- left_join(Days, GaringerOzone.2)
```

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

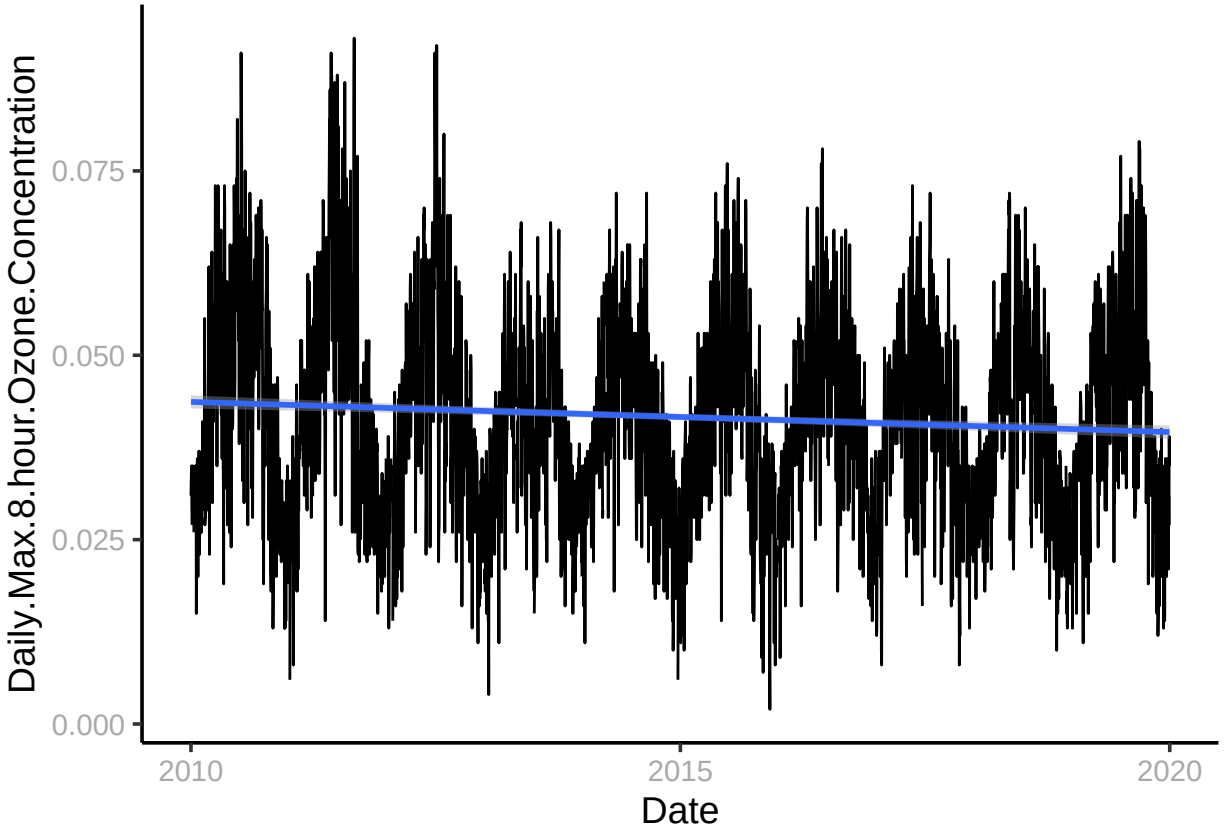
Visualize

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(GaringerOzone, aes(x=Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method="lm")
```

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

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



Answer:Ozone concentration looks to be decreasing - slightly - overall, over the entire time frame. It also looks to have some kind of seasonality that accounts for much of the noise within each year on the plot, as we see a repeated up and down pattern each year.

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
GaringerOzone.3 <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(DAILY_AQI_VALUE = zoo::na.approx(DAILY_AQI_VALUE))

summary(GaringerOzone.3)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151                      Mean   : 41.41
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
```

```
#NAs are gone
```

Answer: In this data, it looks as if we have seasonality within our larger trend. We don't want

to use a piecewise interpolation because this would fill in missing values with the values of the 'nearest neighbors' in the dataset. Because of the variation in our data and because we are looking at the patterns in this variation, we do not want to fill in missing data points with values that are equal to nearby data points. The seasonality in this data looks to increase and decrease somewhat linearly, so it would make most sense for us to use a linear interpolation to connect our points. A spline interpolation is not needed here because the data more closely follow patterns with straight lines than those from the quadratic equation. Using a linear interpolation best allows us to fill in the data with approximations that match the seasonality within the dataset.

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 <- GaringerOzone.3 %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  select(Daily.Max.8.hour.Ozone.Concentration, Date) %>%
  group_by(Date) %>%
  summarise(mean.oz = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

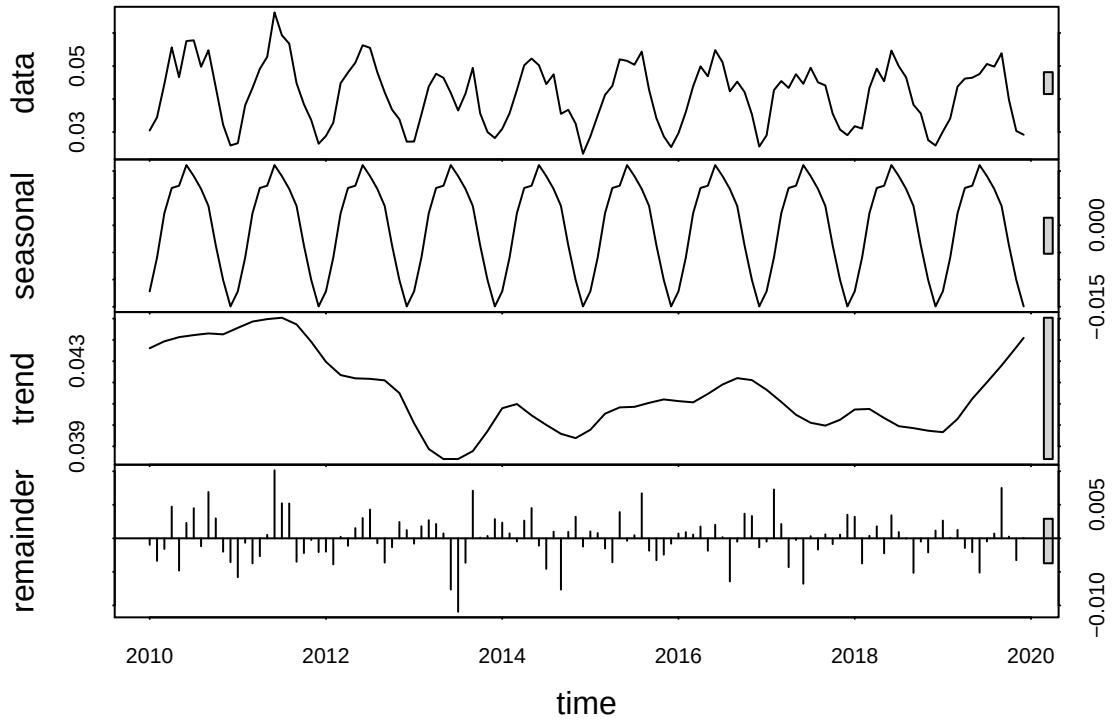
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10
GaringerOzone.daily <-
  GaringerOzone.3 %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration)

fmonth <- month(first(GaringerOzone.daily$Date))
fyear <- year(first(GaringerOzone.daily$Date))

#lday <- day(last(GaringerOzone.daily$Date))
#lmonth <- month(last(GaringerOzone.daily$Date))
#lyear <- year(last(GaringerOzone.daily$Date))

#R should automatically stop at the most recent date in the dataset - specifying end
#dates makes the daily TS too short - I discussed with Luana and Abhishek but can't
#figure out what's going on. I left in my code for lday, lmonth, and lyear which I
#would include in the same manner as start below, so you can see what I attemped to
#do. But it makes the TS only 3297 rather than 3652 elements long.

GaringerOzone.daily_ts <- ts(GaringerOzone.daily$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(fyear, fmonth),frequency = 365)

fmonth2 <- month(first(GaringerOzone.monthly$Date))
fyear2 <- year(first(GaringerOzone.monthly$Date))
```

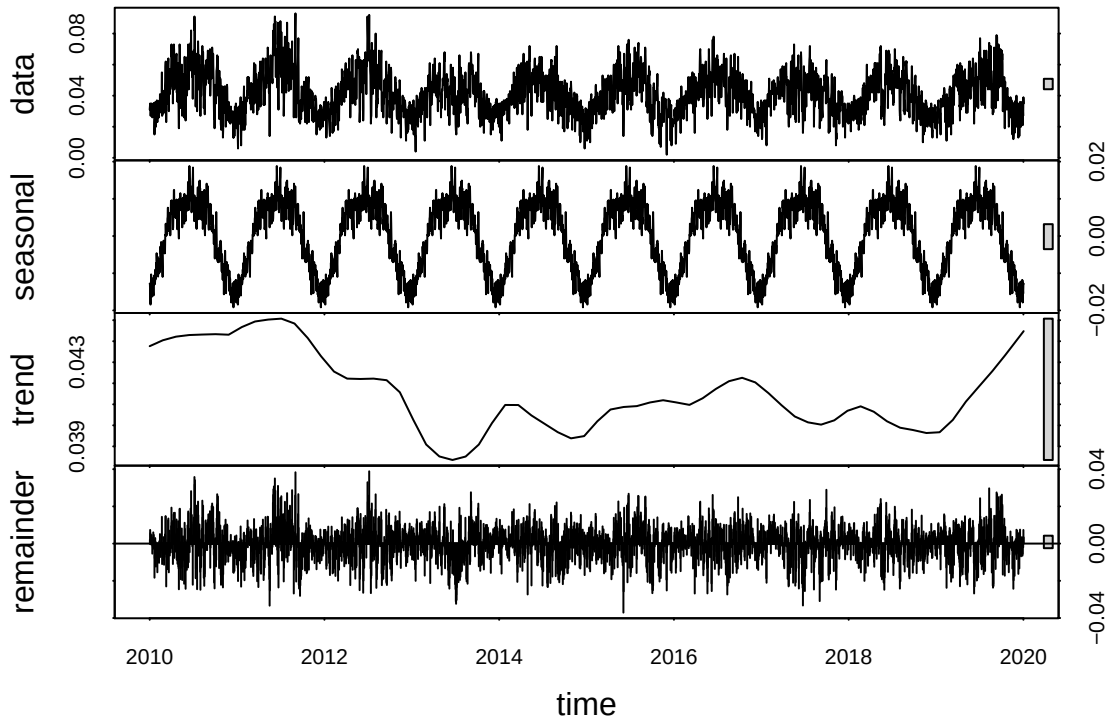
```
lmonth2 <- month(last(GaringerOzone.monthly$Date))
lyear2 <- year(last(GaringerOzone.monthly$Date))

GaringerOzone.monthly_ts <- ts(GaringerOzone.monthly$mean.oz,
                               start = c(fyear2, fmonth2), end = c(lyear2, lmonth2),
                               frequency = 12)
```

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

```
#11
month.decomp <- stl(GaringerOzone.monthly_ts, s.window = "periodic")
plot(month.decomp)
```




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

mo.tr <- Kendall::SeasonalMannKendall(GaringerOzone.monthly_ts)
summary(mo.tr)
```

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

```
#reject H0 bc p < 0.05 --> this data is not stationary - there is a trend in it.

day.tr <- Kendall::SeasonalMannKendall(GaringerOzone.daily_ts)
summary(day.tr)
```

```
## Score =  -739 , Var(Score) = 45223.67
## denominator =  16213.86
## tau = -0.0456, 2-sided pvalue =0.00051075
```

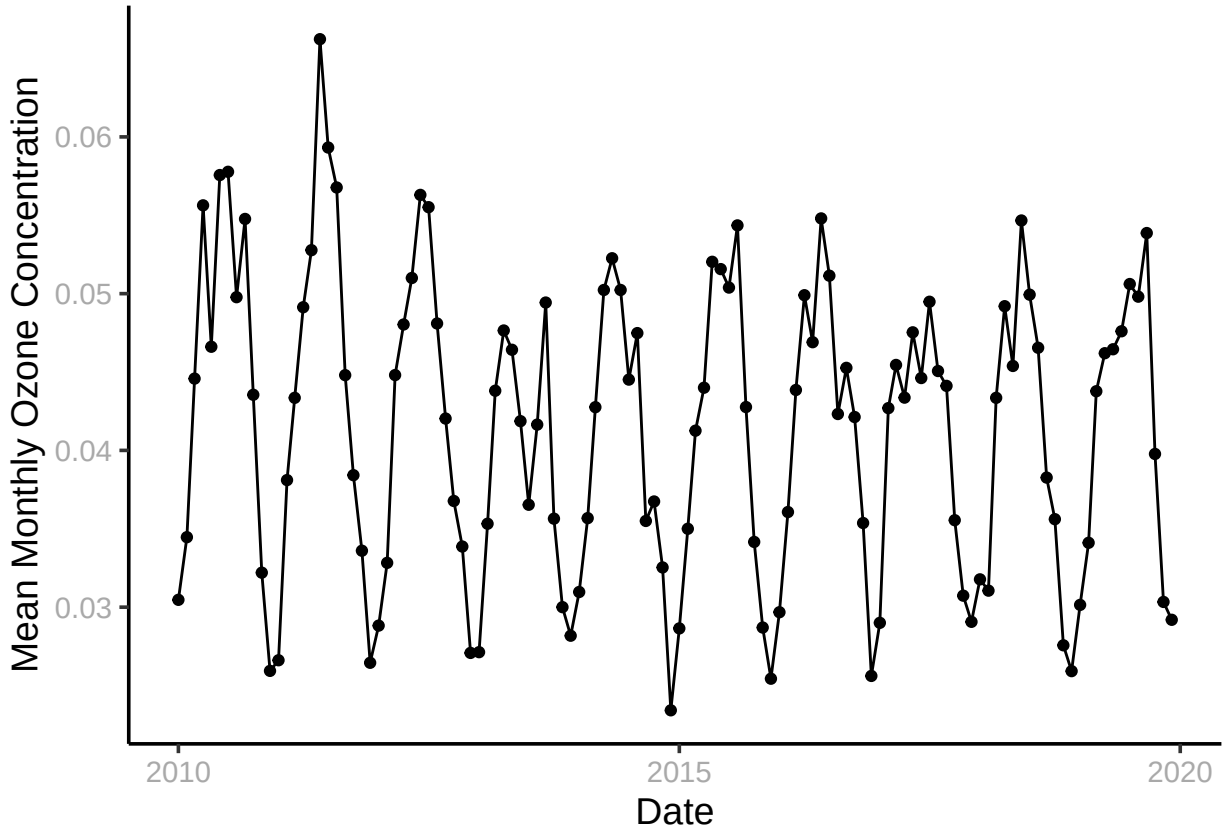
```
#reject H0 - data is not stationary - there is a trend.
```

Answer: Seasonal Mann-Kendall is most appropriate here based on the plots we created from the decomposed time series above. There is a clear seasonality pattern in both the daily and monthly time series, and Seasonal Mann-Kendall is the only trend analysis that can be used on time series with a seasonal component to them.

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
monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean.oz)) +
```

```
  geom_point() +
  geom_line() +
  xlab("Date") +
  ylab("Mean Monthly Ozone Concentration")
print(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: When looking at this graph, we can discern a slight negative trend in mean monthly ozone concentration over full dataset from 2010 to 2020, which is supported by the Seasonal Mann-Kendall test (p = 0.046, tau = -0.143). A negative tau suggests a negative relationship, and our null hypothesis - that there is no trend in the data (or, the data is stationary), is rejected based on our p-value falling below our alpha level.

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
Monthly.Components <- as.data.frame(month.decomp$time.series[,1:3])

Monthly.new <- Monthly.Components %>%
  mutate(mean.oz = trend + remainder) %>%
  select(mean.oz)
```

```r
Monthly.new$Date <- GaringerOzone.monthly$Date

mo.new_ts <- ts(Monthly.new$mean.oz, start = c(fyear2, fmonth2), frequency = 12)

#16

mo.tr2 <- Kendall::MannKendall(mo.new_ts)
summary(mo.tr2)
```

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

```r
#reject H0, there is a trend.
```

Answer: In the initial monthly time series' seasonal trend test, we saw a negative trend (tau= -0.143) and a significant p-value (p = 0.046). With the seasonal data removed, our new time series' trend test also shows a significan p-value and a negative trend. In this second test, our tau is slightly larger (in absolute value), indicating a slightly stronger trend (tau = -0.165). Our p-value for this trend test is also stronger (p = 0.007), as it is clearly less than alpha (a = 0.05), whereas the first trend toes the line of significance. The first p-value is significant, but if rounded up, it would not be, while the second p-value is much less than 0.05. But overall, both the seasonal and normal Mann-Kendall tests point to a significantly decreasing trend in mean monthly ozone concentration in the study area from 2010 to 2020.