Assignment 8: Time Series Analysis

Laura Brockington

Spring 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

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

```
#1
library(here); library(tidyverse); library(lubridate)
library(zoo); library(trend) #loading packages

here() #checking working directory
```

```
## [1] "/Users/laura/Desktop/EDA/EDA"
```

```
mytheme <- theme_gray(base_size = 12) +
  theme(axis.text = element_text(family = "serif",
                                 color = "darkgoldenrod"),
        axis.title = element_text(family = "serif",
                                  color = "chocolate4"),
        axis.ticks = element_line(color = "darkgoldenrod",
                                  linewidth = 0.3),
        plot.title = element_text(family = "serif",
                                  face = "bold",
```

```
                                    color = "chocolate4",
                                    hjust = 0.5),
        panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(color = "darkgoldenrod",
                                        linetype = "solid",
                                        linewidth = 0.3),
        panel.grid.minor = element_line(color = "darkgoldenrod2",
                                        linetype = "dashed")) #creating a ggplot plot theme
theme_set(mytheme) #setting theme as my default
```

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
air_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                     stringsAsFactors = TRUE)
air_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                     stringsAsFactors = TRUE)
air_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                     stringsAsFactors = TRUE)
air_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                     stringsAsFactors = TRUE)
air_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                     stringsAsFactors = TRUE)
air_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                     stringsAsFactors = TRUE)
air_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                     stringsAsFactors = TRUE)
air_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                     stringsAsFactors = TRUE)
air_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                     stringsAsFactors = TRUE)
air_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                     stringsAsFactors = TRUE) #loading in all ten datasets

GaringerOzone <- bind_rows(air_2010, air_2011, air_2012, air_2013, air_2014,
                           air_2015, air_2016, air_2017, air_2018, air_2019)
#combining datasets into one dataframe
```

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$Date <- mdy(GaringerOzone$Date)
#setting data column as date class

#4
GaringerOzone <- #wrangling data to only included 3 columns
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5
Days <- as.data.frame(seq(ymd("2010-01-01"), ymd("2019-12-31"), by = "days"))
colnames(Days) <- c("Date") #creating dataframe of date sequence

#6
GaringerOzone <- left_join(Days, GaringerOzone)
```

```
## Joining with `by = join_by(Date)`
```

```
#combining dataframes to have all days included
```

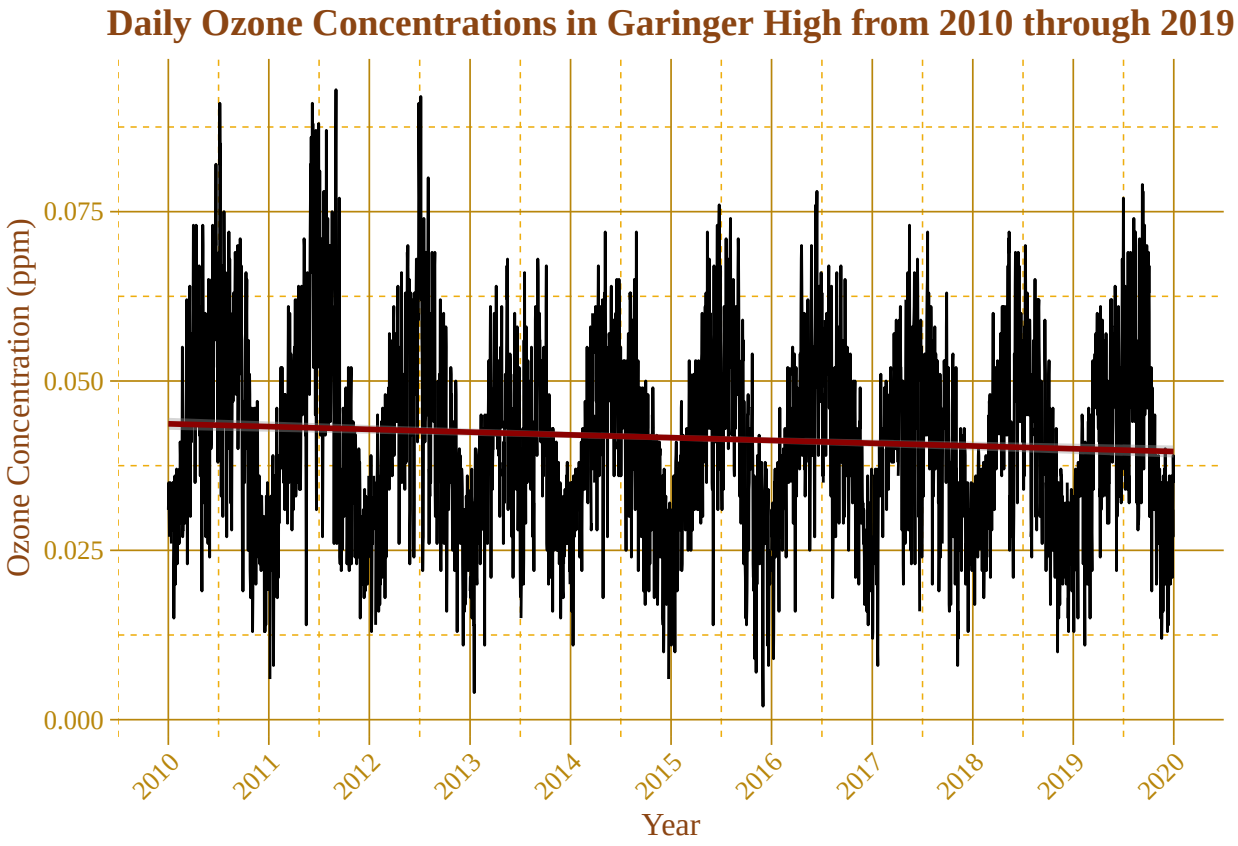
Visualize

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
GaringerOzone.plot <- #plotting ozone concetration over time
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  ylab("Ozone Concentration (ppm)") +
  xlab("Year") +
  geom_smooth(method = lm, col = "darkred") +
  ggtitle("Daily Ozone Concentrations in Garinger High from 2010 through 2019") +
  scale_x_date(breaks = "years", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(GaringerOzone.plot)
```

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

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



Answer: There is a slight negative slope to our trend line, suggesting a possible negative trend in ozone concentration over time. More research is needed to confirm this.

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 <- #using linear interpolation to replace missing values
  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 used a linear interpolation because (1) it is the most straightforward in that it just connects the already existing data points, (2) It doesn't alter the portraial of the existing data, and (3) it's typically the most used method of interpolation for time series analysis.

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 <- #wrangling data to have monthly averages
  GaringerOzone %>%
  separate(Date, c("Year","Month", "Null")) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  group_by(Year, Month) %>%
  summarize(Mean.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  select(Year, Month, Mean.Ozone.Concentration)
```

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

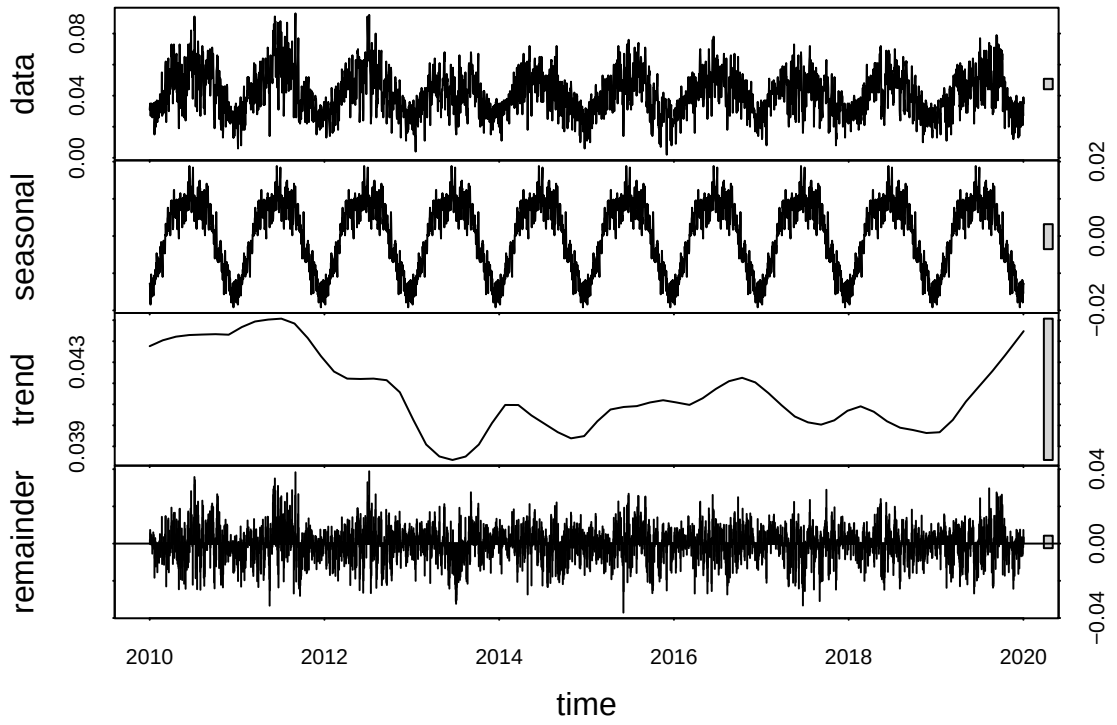
```
GaringerOzone.monthly <- #adding year-month-day column for each month
  GaringerOzone.monthly %>%
  mutate(Date = make_date(year = Year, month = Month, day = 01)) %>%
  select(Date, Year, Month, Mean.Ozone.Concentration)
```

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.

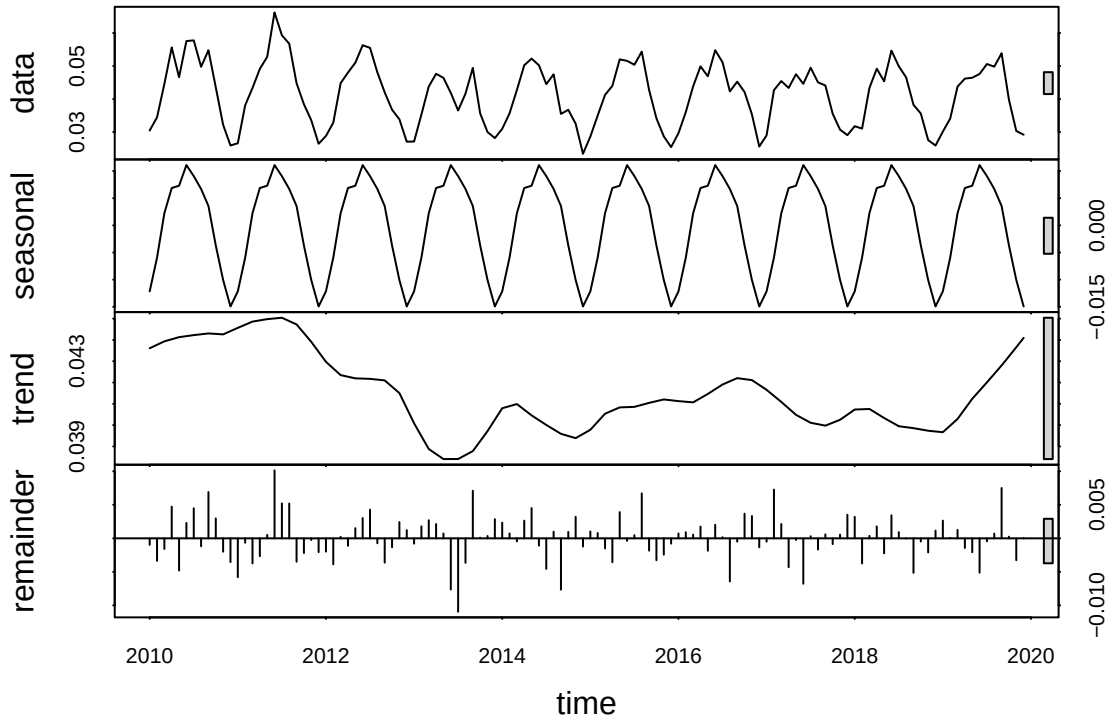
```
#10
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))
f_day <- day(first(GaringerOzone$Date))
#creating object of first month, year, and day

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(f_year, f_month, f_day),
                             frequency = 365) #creating daily time series object

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Ozone.Concentration,
                               start = c(f_year, f_month),
                               frequency = 12) #creating monthly time series object
```

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

```
#11
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
#decomposing the daily time series
plot(GaringerOzone.daily.decomposed) #plotting the components
```




```
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
#decomposing the monthly time series
plot(GaringerOzone.monthly.decomposed) #plotting the components
```



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
GaringerOzone.monthly.trend <- trend::smk.test(GaringerOzone.monthly.ts)
#running a seasonal Mann-Kendall monotonic trend analysis on the monthly ozone series
summary(GaringerOzone.monthly.trend)
```

```
##
##  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
```

```
GaringerOzone.monthly.trend
```

```
##
##  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
```

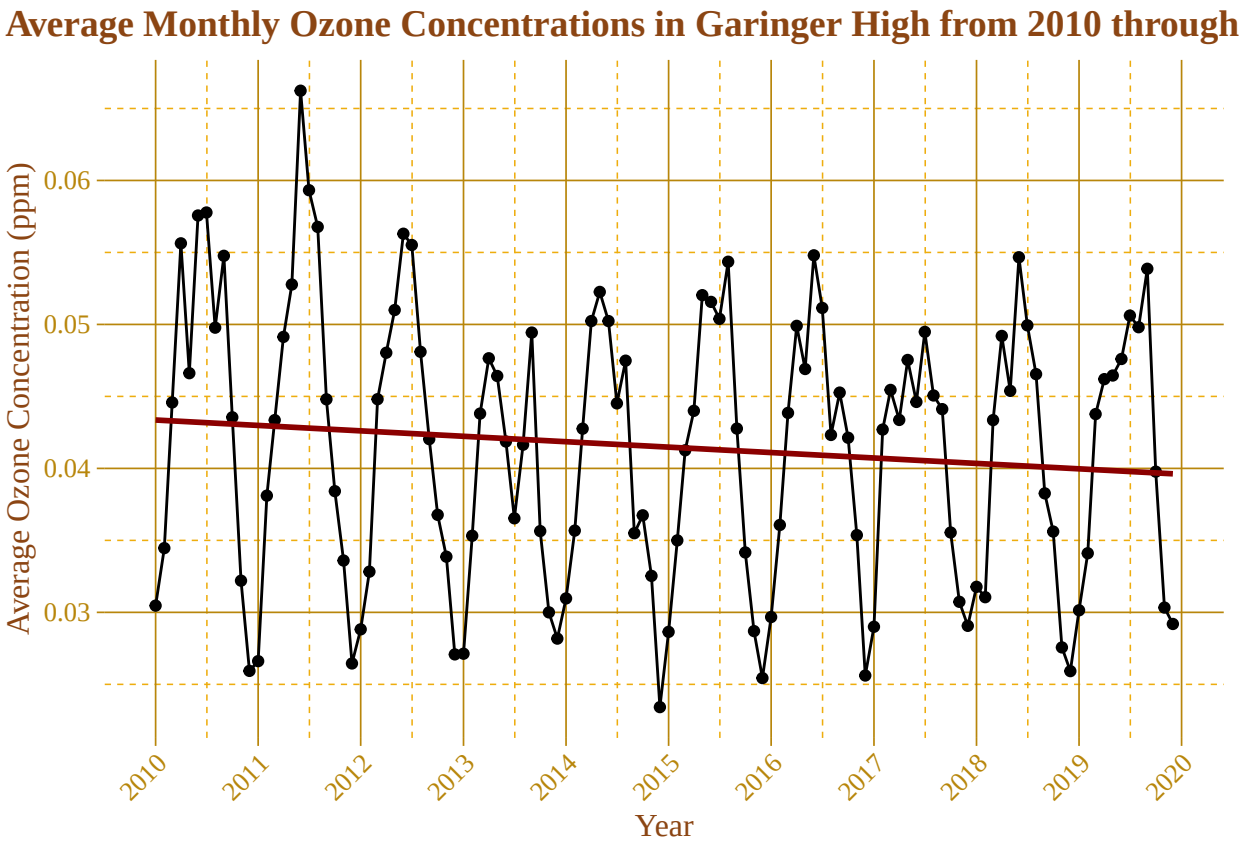
Answer: Our ozone concentration data has seasonality and therefore the SMK is the only appropriate trend analysis, unless we remove the seasonality first.

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
GaringerOzone.monthly.plot <- #plotting monthly ozone concentration over time
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Ozone.Concentration)) +
  geom_line() +
```

```
  geom_point() +
  ylab("Average Ozone Concentration (ppm)") +
  xlab("Year") +
  scale_x_date(breaks = "years", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_smooth(method = lm, se = F, col = "darkred") +
  ggtitle("Average Monthly Ozone Concentrations in Garinger High from 2010 through 2019")
print(GaringerOzone.monthly.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: Average monthly ozone concentrations at Garinger High School have changed in the 2010s, with a statistically significant decrease in ozone from 2010 to 2019 (S=-77, p-value=0.4965).

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
#Extracting all components and adding together the trend and remainder
GaringerOzone.monthly.components <-
  as.data.frame(GaringerOzone.monthly.decomposed$time.series[,2:3])

GaringerOzone.monthly.nonseasonal <-
  mutate(GaringerOzone.monthly.components,
         Non_seasonal = trend + remainder,
         Date = GaringerOzone.monthly$Date)

#16
GaringerOzone.nonseasonal.ts <-
  ts(GaringerOzone.monthly.nonseasonal$Non_seasonal,
                               start = c(f_year, f_month),
                               frequency = 12) #creating time series object

#running non-seasonal Mann-Kendall on monthly average after removing seasonality
GaringerOzone.monthly.nonseasonal.trend <-
  trend::mk.test(GaringerOzone.nonseasonal.ts)
summary(GaringerOzone.monthly.nonseasonal.trend)
```

```
##             Length Class  Mode
## data.name   1      -none- character
## p.value     1      -none- numeric
## statistic   1      -none- numeric
## null.value  1      -none- numeric
## parameter   1      -none- numeric
## estimates   3      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## pvalg       1      -none- numeric
```

```
GaringerOzone.monthly.nonseasonal.trend
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.nonseasonal.ts
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

Answer: After removing seasonality from our monthly ozone concentrations, we found a much larger and more significant decrease in average monthly ozone concentration from 2010 through 2019 (S=-1179, p-value=0.00754).