Assignment 8: Time Series Analysis

Gaby Czarniak

Fall 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

```
getwd()
```

```
## [1] "/home/guest/gaby-cz_EDE_Fall2023"
```

```
library(tidyverse)
library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
library(dplyr)

# Build ggplot theme
```

```
gctheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        # setting base format for legend
        legend.position = "right",
        legend.justification = "left",
        legend.title.align = 0)
# Set as default theme
theme_set(gctheme)
```

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
#Read Ozone data
Ozone_2010_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2011_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2012_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2013_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2014_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2015_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2016_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2017_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2018_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = TRUE)
Ozone_2019_data <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                         stringsAsFactors = TRUE)

# Combine datasets into a single dataframe
GaringerOzone <- rbind(Ozone_2010_data, Ozone_2011_data, Ozone_2012_data,
                       Ozone_2013_data, Ozone_2014_data, Ozone_2015_data,
                       Ozone_2016_data, Ozone_2017_data, Ozone_2018_data,
                       Ozone_2019_data)
```

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 column as date class
is.Date(GaringerOzone$Date)
```

```
## [1] FALSE
```

```
# Column is not a date; it's a factor
GaringerOzone$Date <- as.Date(
  GaringerOzone$Date , format = "%m/%d/%Y")
is.Date(GaringerOzone$Date) #true
```

```
## [1] TRUE
```

```
# 4
# Wrangle data set
GaringerOzone <- GaringerOzone %>%
  # Reduce what columns are present
  select(
    Date, Daily.Max.8.hour.Ozone.Concentration,
    DAILY_AQI_VALUE)

# 5
# Generate new daily dataset
# Fill missing days with NA
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"),
                               to = as.Date("2019-12-31"), by = "day"))
# rename columns
colnames(Days) <- "Date"

# 6
# Combine data frames
GaringerOzone <- left_join(Days, GaringerOzone, by = c("Date"))
```

Visualize

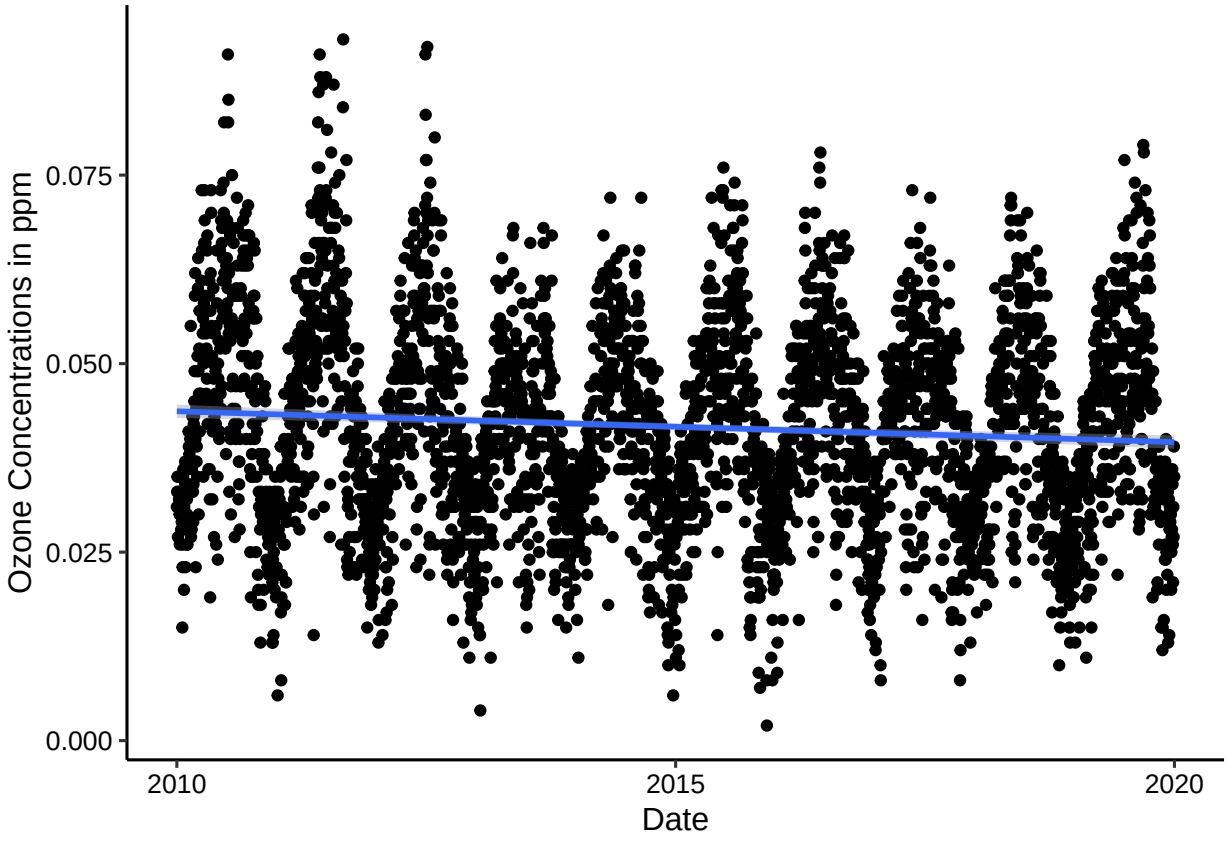
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```r
#7
Ozone_plot <- ggplot(GaringerOzone,
                     aes(x=Date,
                         y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  ylab("Ozone Concentrations in ppm")
print(Ozone_plot)
```

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

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

```
## Warning: Removed 63 rows containing missing values (`geom_point()`).
```



Answer: My plot does not show a super clear suggestion of a trend in ozone concentration over time–it looks like a very slight decrease, but I would want to run more analyses to confirm.

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
head(GaringerOzone)
```

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                   NA              NA
```

```
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
# Adding new column with no missing obs
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration)

summary(GaringerOzone)
```

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

Answer: We didn't use a piecewise function because it would have made the NAs equal to the nearest neighbors, whether it's an earlier date or a later date, and we didn't use the spline, because we didn't need a quadratic function to fill in what was otherwise falling as a straight line. Because our gaps weren't too long, we used the linear interpolation which functions almost like "connecting the dots; any missing data are assumed to fall between the previous and next measurement, with the newly-interpolated values being added in as a straight line.

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

# change some columns -- add 2 more columns to components data frame
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(
  # add column for year
```

```
  Year = year(Date),
  # add column for month
  Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(mean.ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
#create new Date column with each month-year combo
# set as the first day of the month
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = my(paste0(Month,"-",Year)))
```

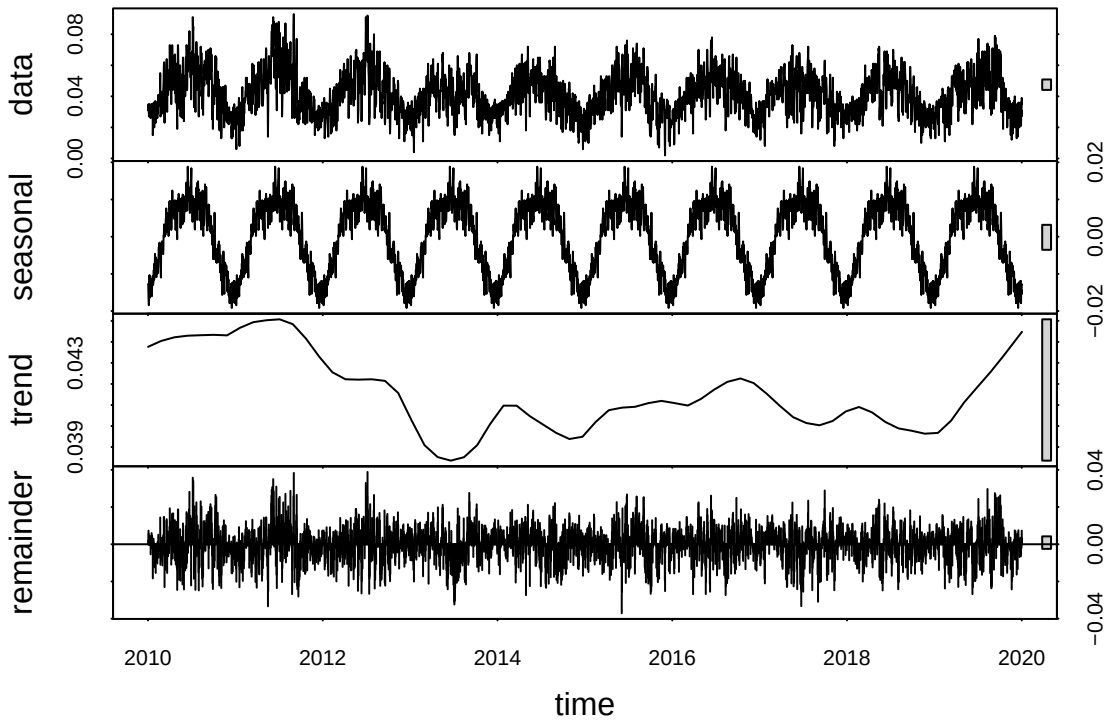
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10
# based on df of daily obs
GaringerOzone.daily.ts <- ts(
                   GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                   start = c(2010,1), frequency = 365)
# plot(GaringerOzone.daily.ts)

# based on monthly avg ozone values
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.ozone,
                   start = c(2010,1), frequency = 12)
# plot(GaringerOzone.monthly.ts)
```

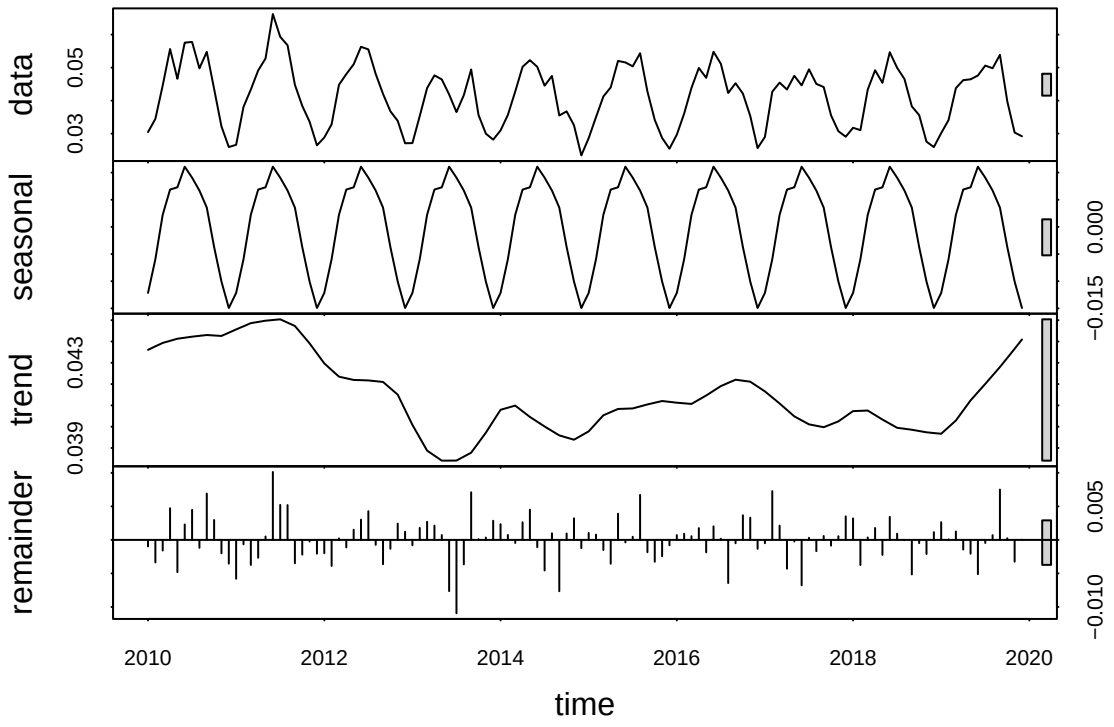
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
# decompose daily
GaringerOzone.daily.decomp <- stl(
  GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```
# seasonal component is much easier to spot

# decompose monthly
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
# Monotonic trend analysis
GaringerOzone.monthly.trend <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
# GaringerOzone.monthly.trend
summary(GaringerOzone.monthly.trend)
```

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

Answer: The seasonal Mann-Kendall is most appropriate because it tests for monotonic trend and can accept seasonality, wheras the regular Mann-Kendall, whose null hypothesis is that the time series is stationary, tests for monotonic trend but cannot be applied to seasonal data.

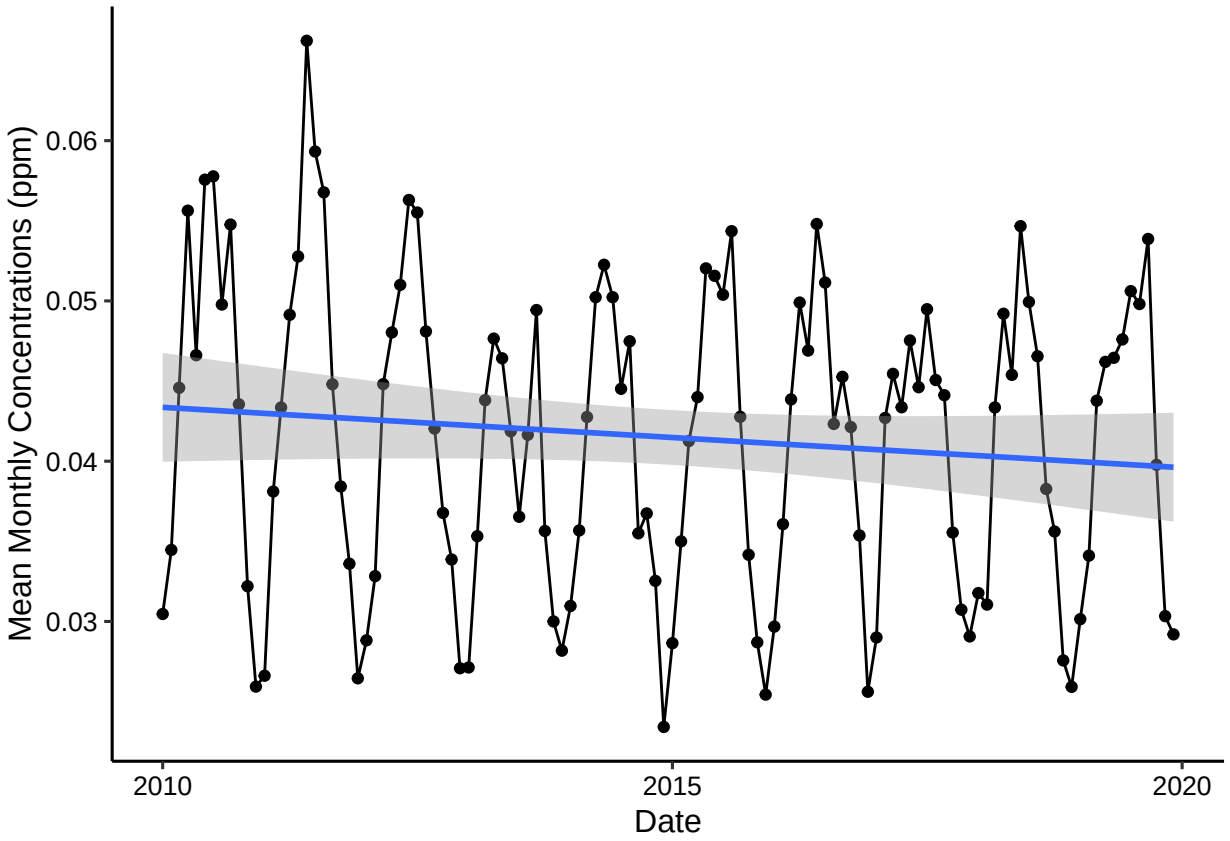
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13

GaringerOzone.monthly.plot <-
```

```r
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean.ozone)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Concentrations (ppm)") +
  geom_smooth( method = lm )
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: Ozone concentrations changed over the 2010s at this station. The results from the seasonal Mann-Kendall are pointing to p value $< .05$ and the null hypothesis for the seasonal Mann-Kendall is that the data is stationary, so we can reject that null hypothesis and say we have a trend (tau = -0.143, 2-sided pvalue =0.046724). There is trend in the data that is not explained by seasonality. Also, when I plotted the components, the grey bars on the right side were extremely small for the seasonal component versus the trend component, leading me to conclude that the seasonal component plays a much smaller role, relative to the trend component, in explaining ozone concentrations over time at this station. Here, in this way, the trend is the most important explanatory component in the dataset.

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 non-seasonal data
GaringerOzone.monthly.components <- as.data.frame(
  GaringerOzone.monthly.decomp$time.series[,1:3])
# plot(GaringerOzone.monthly.components)

# subtract seasonal component
GaringerOzone.subtracted <- GaringerOzone.monthly.components %>%
  mutate(
    Observed = GaringerOzone.monthly$mean.ozone,
    Date = GaringerOzone.monthly$Date,
  ) %>%
  mutate(Nonseasonal = Observed - seasonal)

# create time series object to run test on
GaringerOzone.subtracted.ts <- ts(GaringerOzone.subtracted$Nonseasonal,
                                  start = c(2010,1),
                                  frequency = 12)

#16
#compare with results from part 12 using Mann Kendall
GaringerOzone.monthly.nonseasonal <-
  Kendall::MannKendall(GaringerOzone.subtracted.ts)
# GaringerOzone.monthly.nonseasonal
summary(GaringerOzone.monthly.nonseasonal)
```

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

Answer: Removing the seasonal component and running the nonseasonal MannKendall test returned an even lower p-value and a tau even closer to -1, or an even stronger decreasing slope (tau = -0.165, 2-sided pvalue =0.0075402), which leads me to conclude that removing seasonality showed an even stronger conclusion that ozone concentrations changed over the 2010s at this station (again, there is trend in the data that is not explained by seasonality).