# Assignment 8: Time Series Analysis

Maeve Arthur

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

#check wd
#here()

#loading necessary packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

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

```
library(trend)

#set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

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

#importing datasets
Air_2010 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw

Air_2011 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw

Air_2012 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw

Air_2013 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw

Air_2014 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw

Air_2015 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw

Air_2016 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw

Air_2017 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw

Air_2018 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw

Air_2019 <- read.csv("/home/guest/EDA-Spring2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw

#creating single dataframe from datasets

GaringerOzone <- rbind(Air_2010,Air_2011,Air_2012,Air_2013,Air_2014,Air_2015,Air_2016,Air_2017,Air_2018
```

## 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); class(GaringerOzone$Date)
```

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

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

#5
#defining start date
start_date <- first(GaringerOzone$Date)

#defining end date
end_date <- last(GaringerOzone$Date)

#generating data frame
Days <- as.data.frame(seq(start_date, end_date, "days"))

colnames(Days)[1] = "Date"

#6
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
```

```
## Warning in left_join(Days, GaringerOzone, by = "Date"): Each row in `x` is expected to match at most
## i Row 731 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

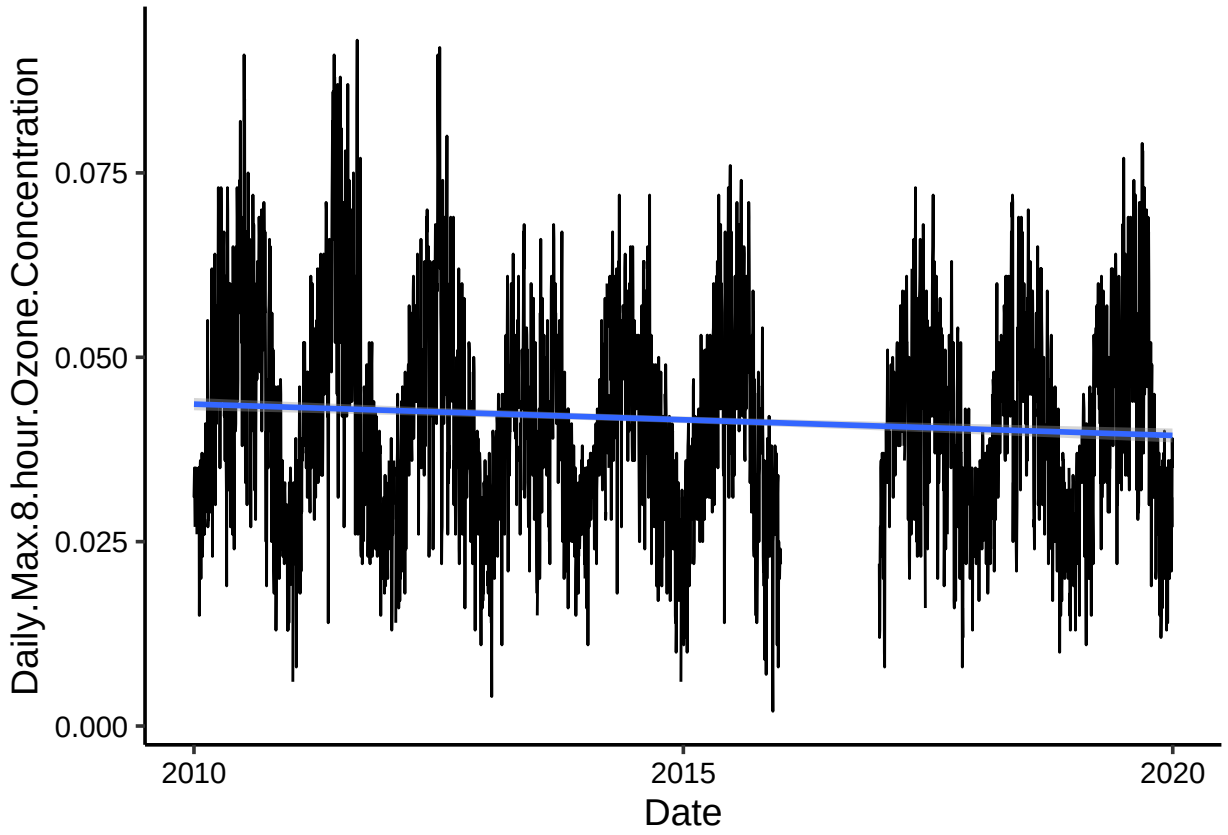
## Visualize

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

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

```
## Warning: Removed 422 rows containing non-finite values (`stat_smooth()`).
```




Answer: The trend line shows a slight decrease in ozone concentratioions 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
GaringerOzone_clean2 <-
  GaringerOzone %>%
  mutate(concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone_clean3 <- GaringerOzone_clean2 %>%
  select(Date, DAILY_AQI_VALUE, concentration.clean)
```

Answer: We used a liner interpolation to fill in missing daily data because our data shows linear trend. If the data had shown a quadratic trend, we would have used spline. Linear interpolation makes the most sense for the data we have.

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_clean3 %>%
  mutate(Month=month(Date),
```

```
        Year=year(Date)) %>%
  group_by(Month,Year) %>%
  summarise(mean.concentration=mean(concentration.clean))
```

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

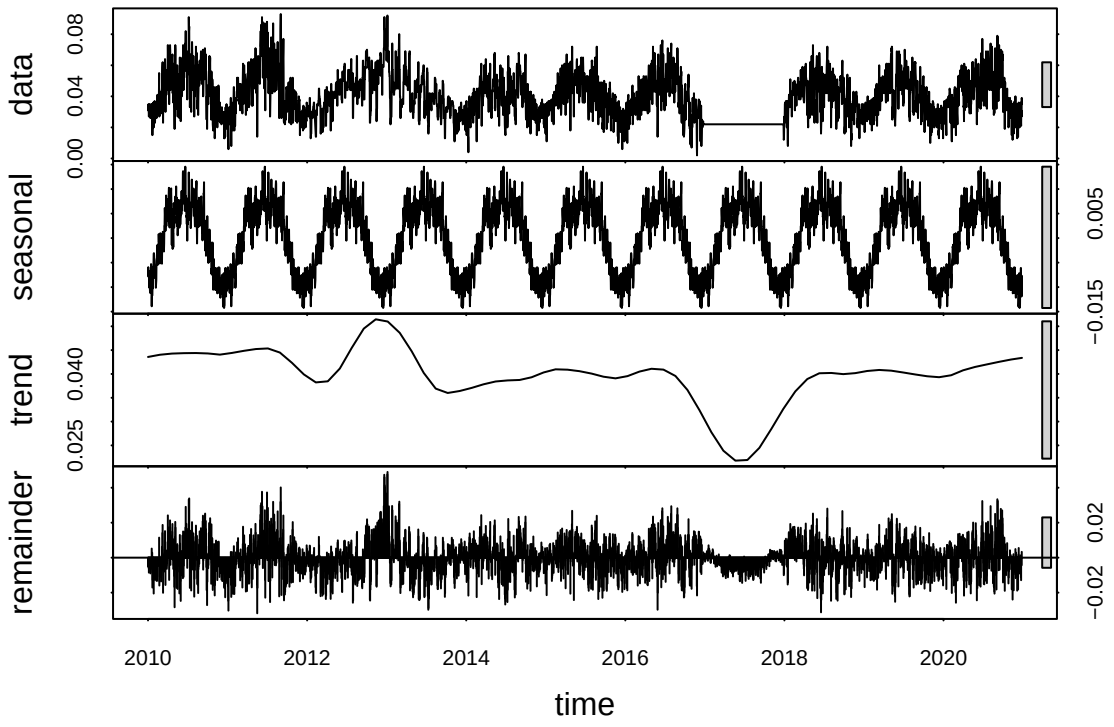
```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date=paste0(Month, "/", 01, "/", Year))

GaringerOzone.monthly$Date <- mdy(GaringerOzone.monthly$Date)
```

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.

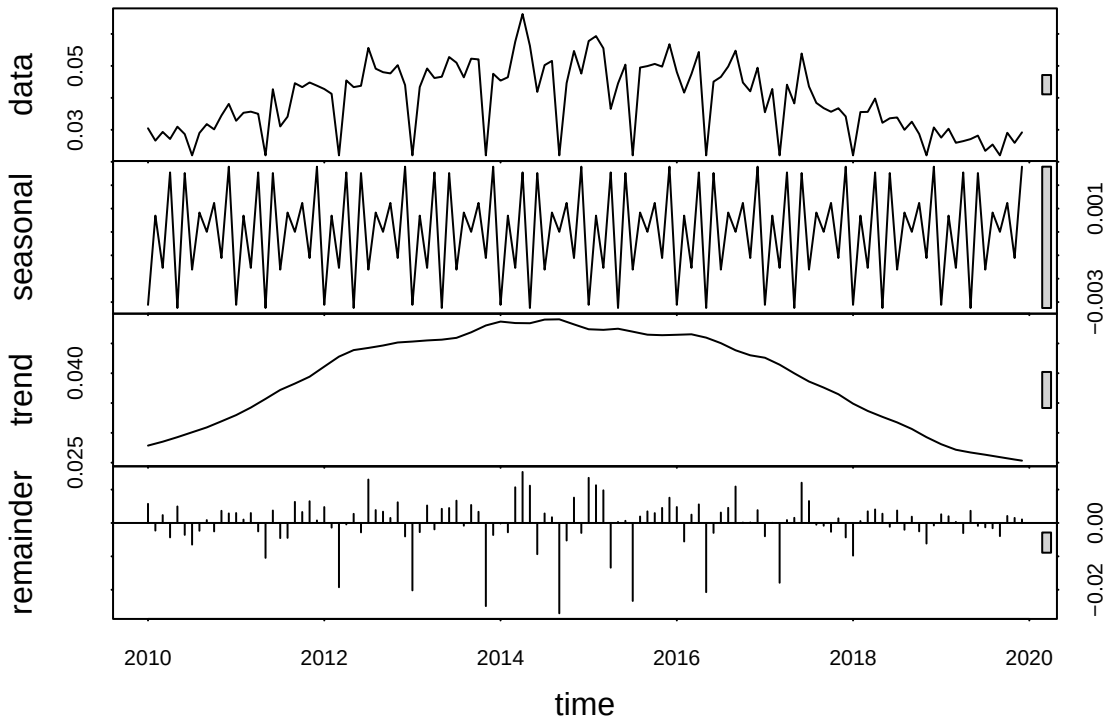
```
#10

#daily ts
f_day <- day(first(GaringerOzone_clean3$Date))
f_month <- month(first(GaringerOzone_clean3$Date))
f_year <- year(first(GaringerOzone_clean3$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone_clean3$concentration.clean,
                   start=c(f_year,f_month,f_day),
                   frequency=365)

#monthly ts
f_month2 <- month(first(GaringerOzone.monthly$Date))
f_year2 <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.concentration,
                   start=c(f_year,f_month),
                   frequency=12)
```

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

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




```
monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(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
monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

monthly_trend
```

```
## tau = -0.0819, 2-sided pvalue =0.25497
```

```
summary(monthly_trend)
```

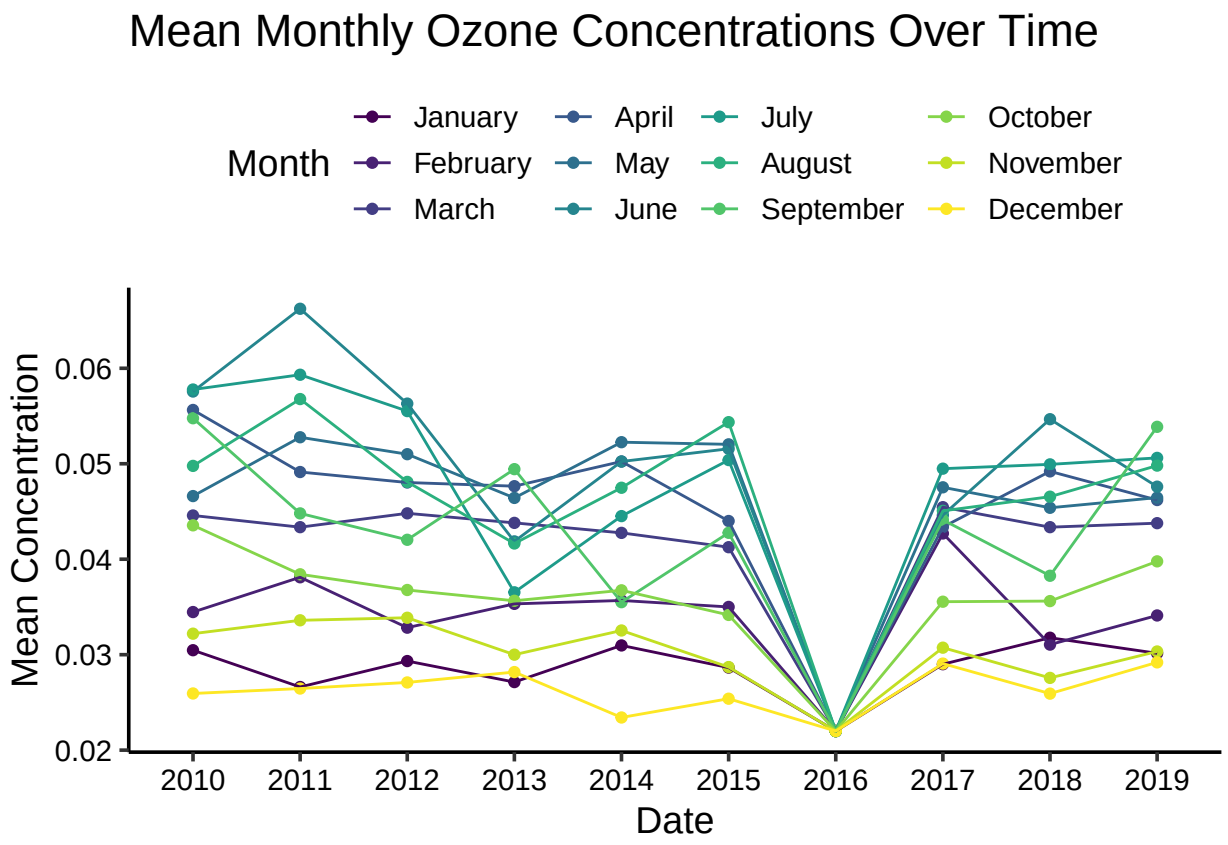
```
## Score =  -44 , Var(Score) = 1494
## denominator =  536.9832
## tau = -0.0819, 2-sided pvalue =0.25497
```

Answer: The seasoonal Mann-Kendall is appropriate here because the decomposition plot clearly shoows that ozone concentration follows a seasonal trend.

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$Month <- month((GaringerOzone.monthly$Month), label = TRUE, abbr = FALSE)

monthly_plot <-
ggplot(GaringerOzone.monthly, aes(
  x = factor(Year),
  y = mean.concentration,
```

```
  color=as.factor(Month))) +
  geom_point() +
  geom_line(aes(group=Month)) +
  labs(
    title="Mean Monthly Ozone Concentrations Over Time",
    x = "Date",
    y = "Mean Concentration",
    color = "Month") +
  mytheme

print(monthly_plot)
```



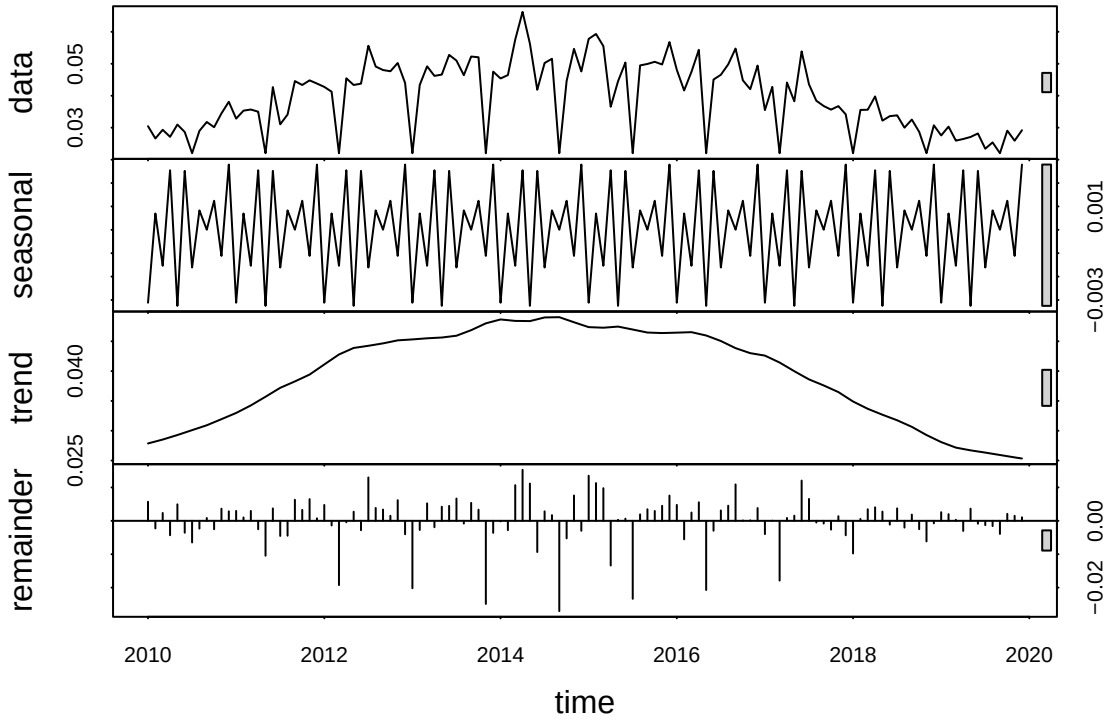

14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: As the lines depicting ozone concentration stay relatively stable over time for most months, the plot shows that there has not been a change in ozone concentrations over the 2010s at this station. This is confirmed by the Seasonal Mann-Kendall test, which provides a p-value of greater than 0.05 (p-value = 0.16323).

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
plot(monthly_decomp)
```




```
monthly_components <- as.data.frame(monthly_decomp$time.series[,2:3])

monthly_components <-
  mutate(monthly_components,
         Observed = GaringerOzone.monthly$mean.concentration,
         Date = GaringerOzone.monthly$Date)

#16
non_seasonal_monthly_ts <- ts(monthly_components$Observed,
                   start=c(f_year,f_month),
                   frequency=12)

non_seasonal_trend <- Kendall::MannKendall(non_seasonal_monthly_ts)

summary(non_seasonal_trend)
```

```
## Score =  -650 , Var(Score) = 194152
## denominator =  7105.918
## tau = -0.0915, 2-sided pvalue =0.14078
```

Answer: The Seasonal Mann-Kendall test revealed a p-value of 0.16323, while the Mann-Kendall test revealed a p-value of 0.088483. Neither p-vlaue suggests that there is a significant change in ozone concentrations during this time period. However, the p-value from the Mann-Kendall test is far closer to being siignificant than the Seasonal Mann-Kendall.