Assignment 8: Time Series Analysis

Desa Bolger

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

```
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2023
```

```
here()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(ggplot2)
library(here)
library(ggthemes)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#install.packages("trend")
library(trend)

desa_theme <- theme_base() +
  theme(line = element_line(color='black',linewidth =.5),
        text = element_text(color='black'),
        panel.grid.major =  element_line(color='black', linewidth = .5),
        rect = element_rect(color = 'lightgrey', fill = 'lightgrey'),
        plot.background = element_rect(color = 'lightgrey', fill = 'lightgrey'),
        panel.background =  element_rect(color = 'lightgrey', fill = 'lightgrey'),
        legend.background = element_rect(color='lightblue', fill = 'lightblue'),
        legend.title = element_text(color='darkblue'))

theme_set(desa_theme) #set as 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.

```
#1

NC2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                         stringsAsFactors = TRUE)
```

```
NC2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                         stringsAsFactors = TRUE)
NC2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                         stringsAsFactors = TRUE)
NC2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                         stringsAsFactors = TRUE)
NC2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                         stringsAsFactors = TRUE)
NC2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                         stringsAsFactors = TRUE)
NC2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                         stringsAsFactors = TRUE)
NC2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                         stringsAsFactors = TRUE)
NC2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                         stringsAsFactors = TRUE)
NC2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                         stringsAsFactors = TRUE)
GaringerOzone <- rbind(NC2010, NC2011, NC2012, NC2013, NC2014, NC2015, NC2016,
                       NC2017, NC2018, NC2019)
#view(GaringerOzone)
dim(GaringerOzone)
```

```
## [1] 3589   20
```

```
#dim works 3580 x 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

class(GaringerOzone$Date) #05/27/84 01/01/2010
```

```
## [1] "factor"
```

```
# a factor! we will switch it
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date) #now its a date!
```

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

```
# 4

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

# 5

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

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

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

```
dim(GaringerOzone)
```

```
## [1] 3652    3
```

```
#3652 x 3
```

Visualize

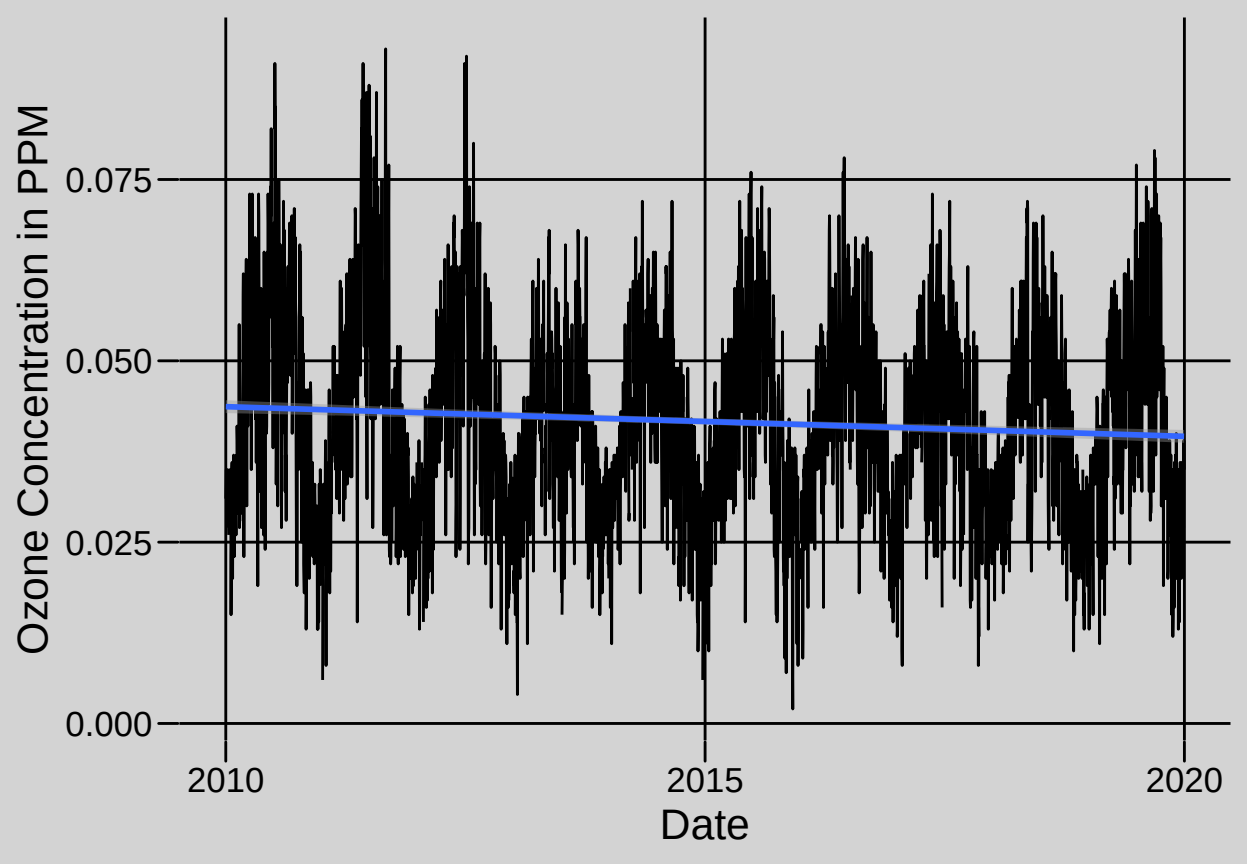
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7

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

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

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



Answer: It suggests a slight decrease 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

GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
#view(GaringerOzone)
```

Answer: We didn't use Spline because that would be quadratic and not linear. We didn't use Piecewise because missing data is assumed to be equal to its neighbor, but we wanted linear because it would draw a straight line to determine missing points through an average of the neighbors.

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)

```r
#9

GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate("Month" = month(Date)) %>%
  mutate("Year" = year(Date)) %>%
  group_by(Year, Month) %>%
  summarize(meanConcentration = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate("Month_Year" = my(paste0(Month, "-", Year)))
```

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

```r
#view(GaringerOzone.monthly)
```

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.

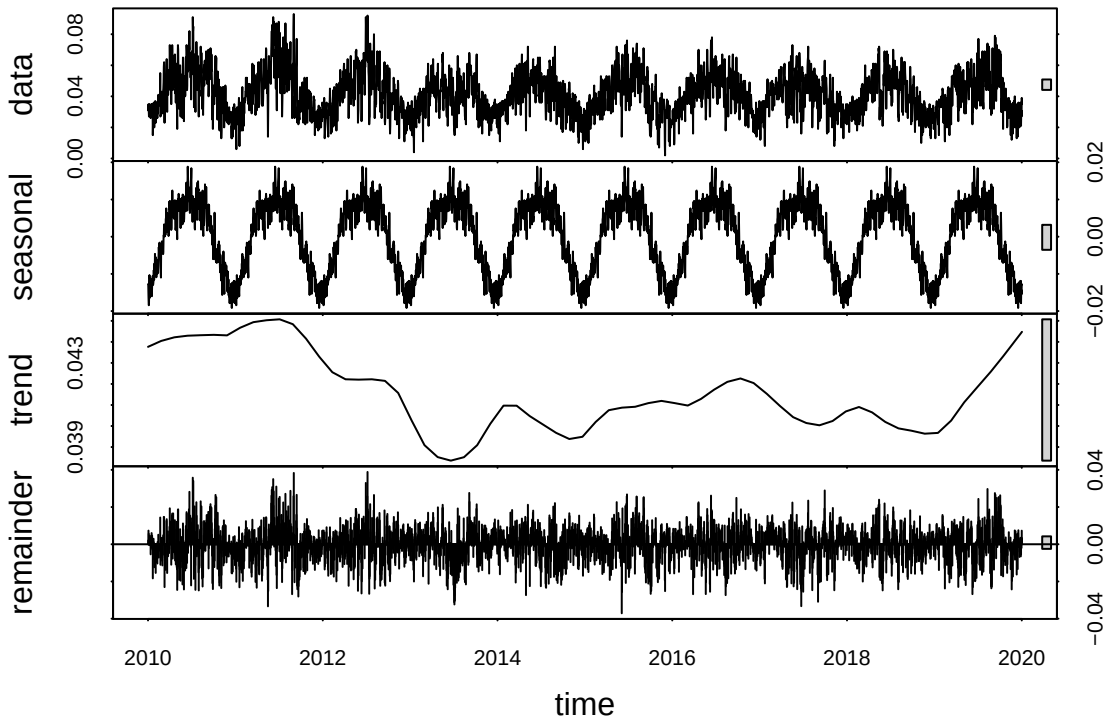
```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                   start= c(2010,1),
                   frequency=365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanConcentration,
                   start= c(2010,1),
                   frequency=12)
```

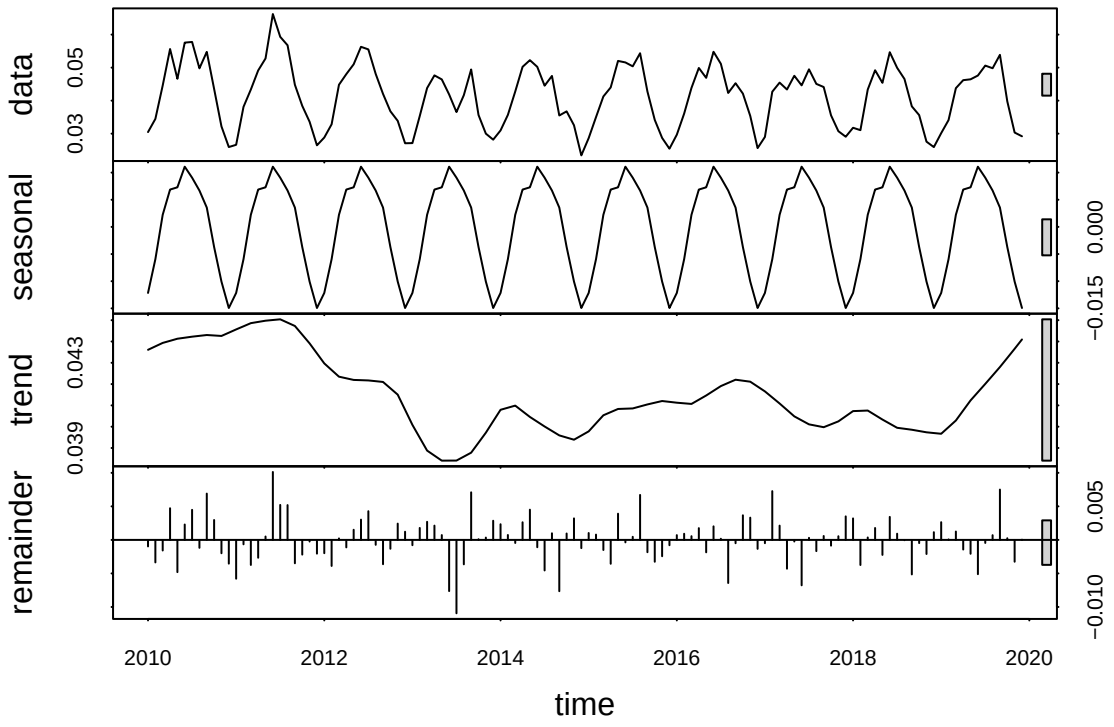
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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

# Visualize the decomposed series.
plot(GaringerOzone.dailyDecomposed)
```




```
GaringerOzone.monthlyDecomposed <- stl(GaringerOzone.monthly.ts,
                                       s.window = "periodic")
# Visualize the decomposed series.
plot(GaringerOzone.monthlyDecomposed)
```



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.monthly1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
GaringerOzone.monthly1
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(GaringerOzone.monthly1)
```

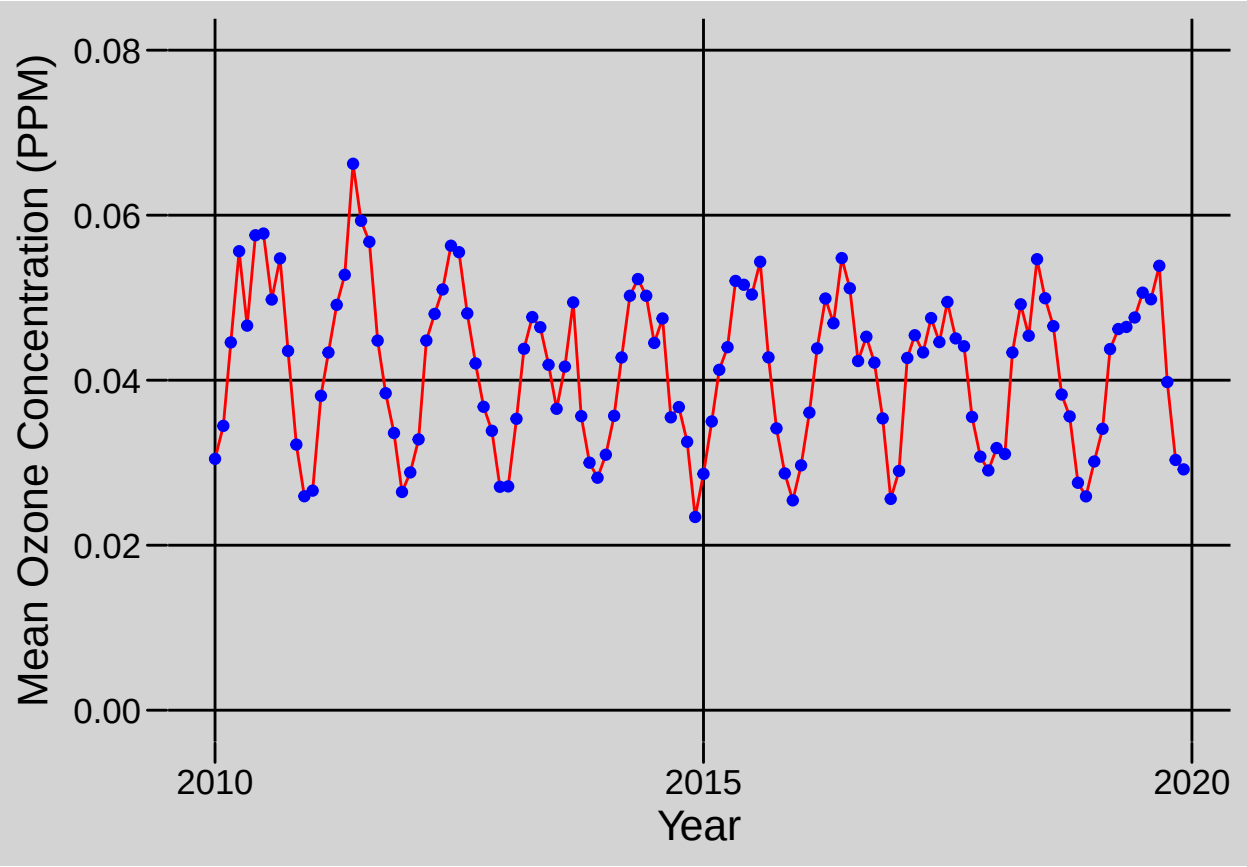
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: We used the seasonal Mann-Kendall because it is clear that there is a seasonal pattern. The regular Mann-Kendall, Spearman Rho, Augmented Dickey Fuller, and linear regression do not do well with 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.

```r
# 13

ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = meanConcentration)) +
  geom_line(color = "red") +
  geom_point(color = "blue") +
  xlab("Year") +
  ylim(0, 0.08) +
  ylab(expression("Mean Ozone Concentration (PPM)"))
```



Have ozone concentrations changed over the 2010s at this station? 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: There is a trend in terms of ozone concentrations over the 2010s (Tau =-0.0143, p-value = 0.046725). We can reject the null hypothesis that there was no change in ozone levels because this test was statistically significant. It seems like there is a negative trend, with PPM decreasing as the year increases.

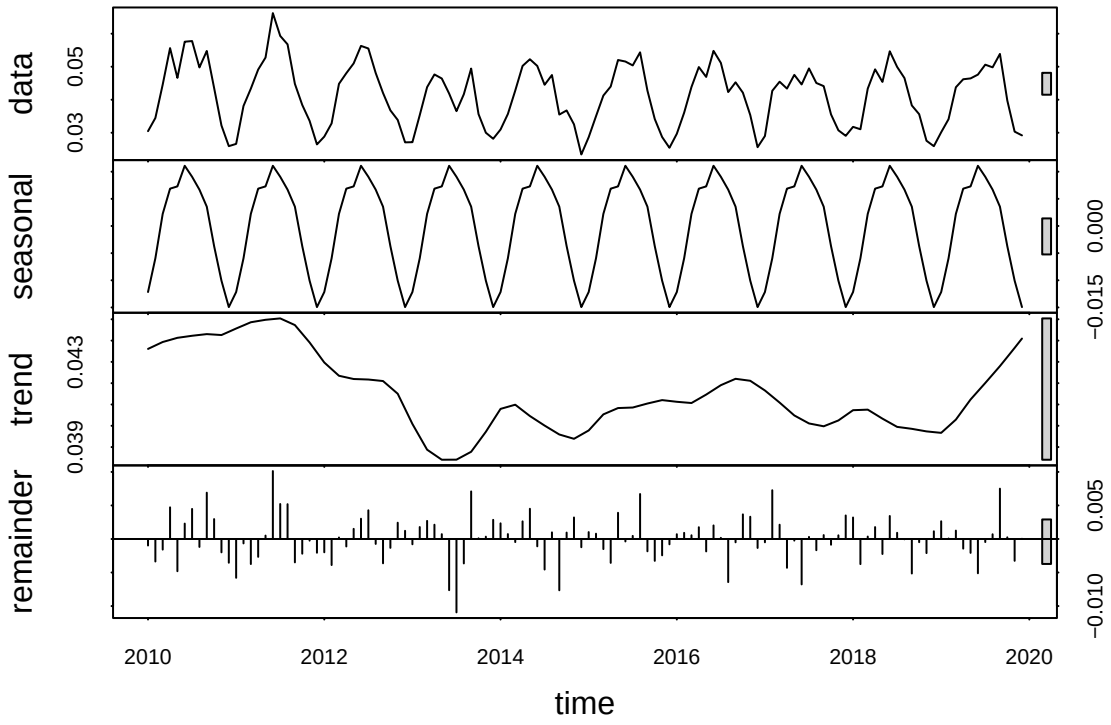
15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

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

# Generate the decomposition
GaringerOzone.monthlyDecomposed <- stl(GaringerOzone.monthly.ts,
                                       s.window = "periodic")

# Visualize the decomposed series.
plot(GaringerOzone.monthlyDecomposed )
```




```
# We can extract the components and turn them into data frames
GaringerOzone.monthlyDecomposed_Components <-
  as.data.frame(GaringerOzone.monthlyDecomposed$time.series[,1:3])

GaringerOzone.monthlyDecomposed_Components <-
  mutate(GaringerOzone.monthlyDecomposed_Components,
         Observed = GaringerOzone.monthly$meanConcentration,
         Date = GaringerOzone.monthly$Month_Year,
         Nonseasonal = Observed - seasonal)

 #view(GaringerOzone.monthlyDecomposed_Components)

#16
a_month <- month(first(GaringerOzone.monthlyDecomposed_Components$Date))
a_year <- year(first(GaringerOzone.monthlyDecomposed_Components$Date))
GO <- ts(GaringerOzone.monthlyDecomposed_Components$Nonseasonal,
```

```
                  start=c(a_year,a_month),
                  frequency=12)

FinalGO <- Kendall::MannKendall(GO)
FinalGO
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(FinalGO)
```

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

Answer: This non-seasonal MannKendall had a tau of -0.165 and a p value of 0.0075402. The seasonal MannKendall had a tau of -0.143 and a p-value of 0.045724.
Only when we use the seasonal MannKendall was our data statistically significant ($p < 0.05$)