Assignment 8: Time Series Analysis

Miaojun Pang

Spring 2024

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] "C:/Users/mpang/OneDrive - Duke University/Desktop/Spring 2023/790 Time Series/EDA_Spring2024_new
```

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v stringr 1.5.0
## v tidyr   1.3.0     v forcats 1.0.0
## v readr   2.1.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
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)
```

```
## Warning: package 'trend' was built under R version 4.2.3
```

```
mytheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
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.

```
#1

library(here)
```

```
## Warning: package 'here' was built under R version 4.2.3
```

```
Data2019<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),stringsAsFactors
Data2018<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFactors
Data2017<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFactors
Data2016<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFactors
Data2015<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFactors
Data2014<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFactors
Data2013<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFactors
Data2012<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFactors
Data2011<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFactors
Data2010<- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFactors

OzoneGH <- rbind (Data2019, Data2018, Data2017, Data2016, Data2015, Data2014, Data2013, Data2012, Data2

dim(OzoneGH)
```

```
## [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

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

# 4

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

# 5

Date<- seq(as.Date('2010-01-01'),
           as.Date('2019-12-31'), by=1)
Days<- as.data.frame(Date)

# 6
OzoneGH<- left_join(Days,OzoneSubset)
```

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

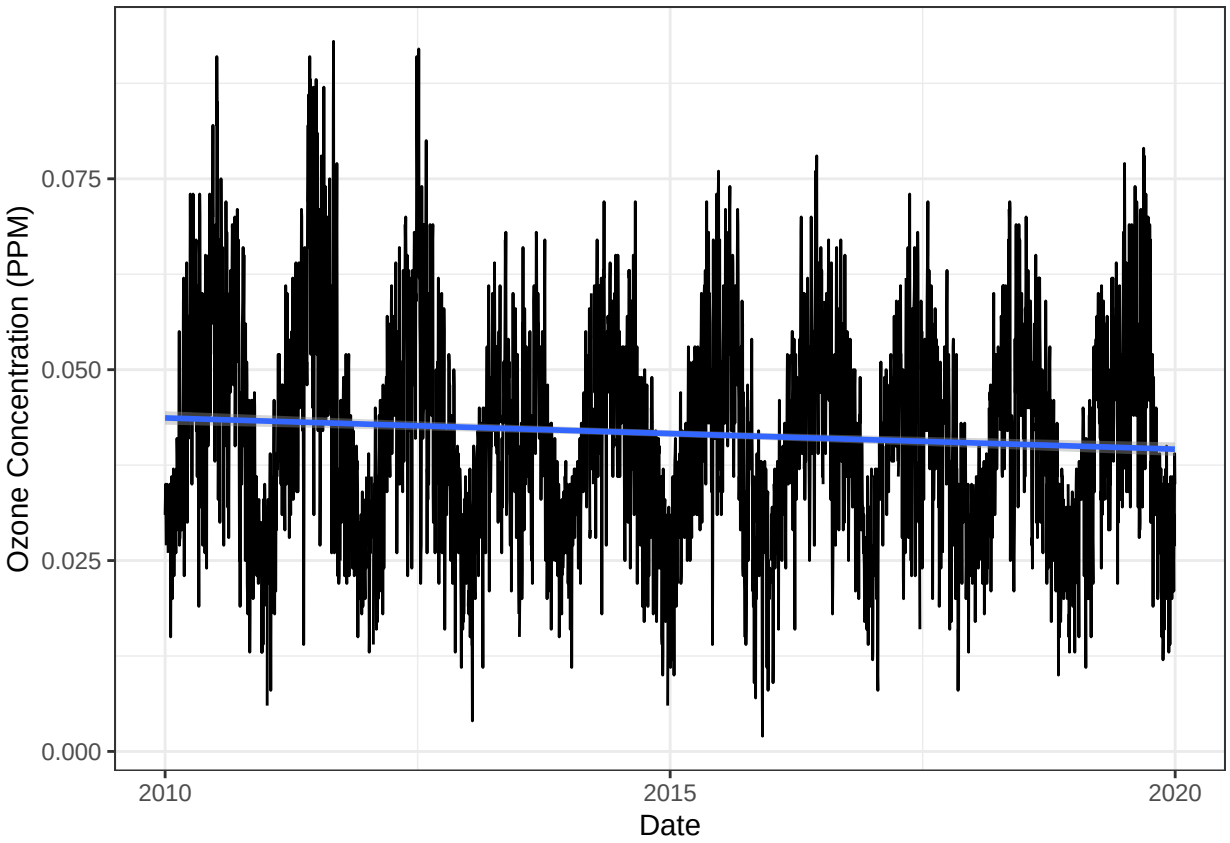
Visualize

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

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

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



Answer: Yes, as the plot suggests a decreasing trend 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

library(zoo)
OzoneGH$Daily.Max.8.hour.Ozone.Concentration<-
  na.approx(OzoneGH$Daily.Max.8.hour.Ozone.Concentration, na.rm = FALSE)
```

Answer: Linear interpolation assumes that the change between two points is linear and fills in missing values accordingly, creating a straight-line connection between known data points. However, when we using a piecewise constant or spline interpolation is based on the characteristics of ozone concentration data and the specific goals of the 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)

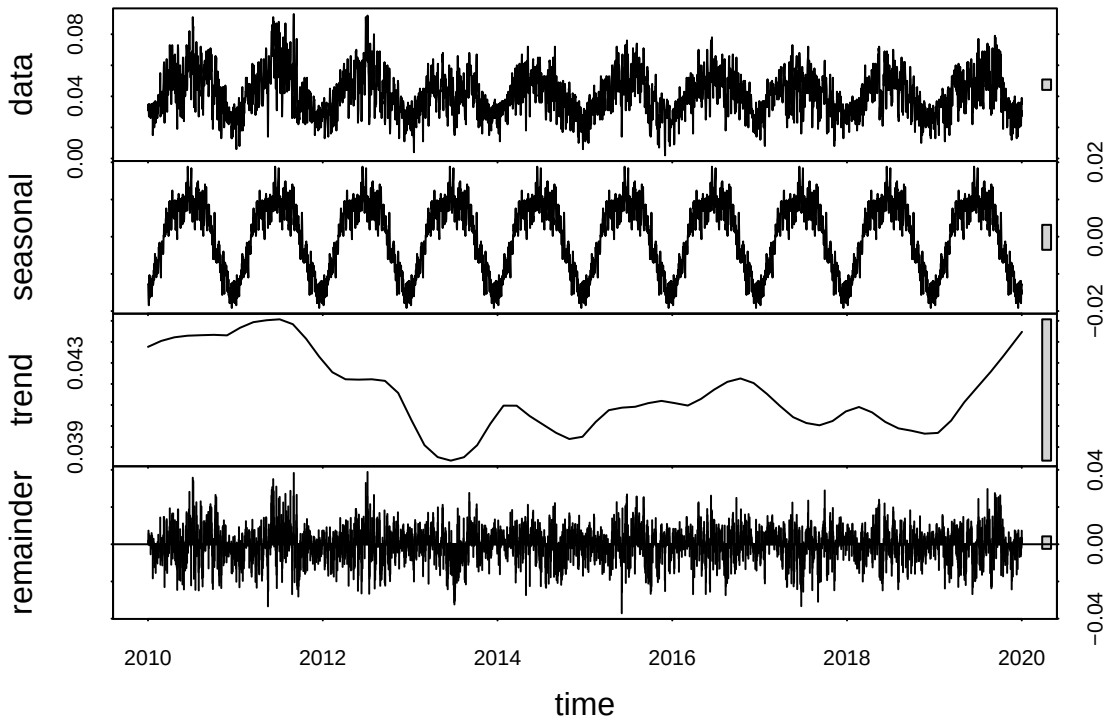
```
#9
OzoneGH.monthly<- OzoneGH%>%
  mutate(month = month(Date),
         year = year(Date))%>%
  mutate(month_year = my(paste0(month, "-", year)))%>%
  dplyr::group_by(month_year)%>%
  dplyr::summarise(MeanMonthlyOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

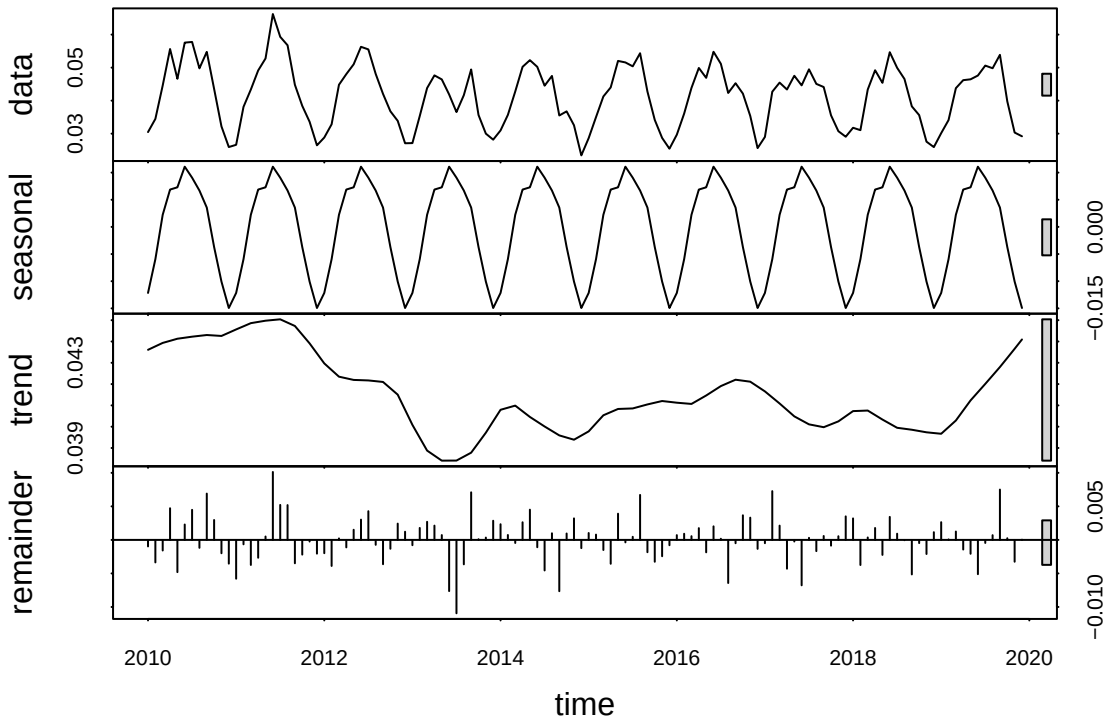
```
#10
firstday<- day(first(OzoneGH$Date))
firstmonth<- month(first(OzoneGH$Date))
firstyear<- year(first(OzoneGH$Date))

OzoneGH.daily.ts<-
  ts(OzoneGH$Daily.Max.8.hour.Ozone.Concentration,
                            start = c(firstyear, firstmonth, firstday),
                            frequency = 365)

FirstMonth<- month(first(OzoneGH.monthly$month_year))
FirstYear<- year(first(OzoneGH.monthly$month_year))

OzoneGH.monthly.ts<-
  ts(OzoneGH.monthly$MeanMonthlyOzone,
     start = c(FirstYear, FirstMonth),
     frequency = 12)
```

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

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




```
OzoneGH.monthly.decompose<-
  stl(OzoneGH.monthly.ts, s.window = "periodic")
plot(OzoneGH.monthly.decompose)
```



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
Ozone.trend<-
  Kendall::SeasonalMannKendall(OzoneGH.monthly.ts)
Ozone.trend
```

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

```
summary(Ozone.trend)
```

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

```
Ozone.trend2 <-
  trend::smk.test(OzoneGH.monthly.ts)

Ozone.trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
```

```
## data:  OzoneGH.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

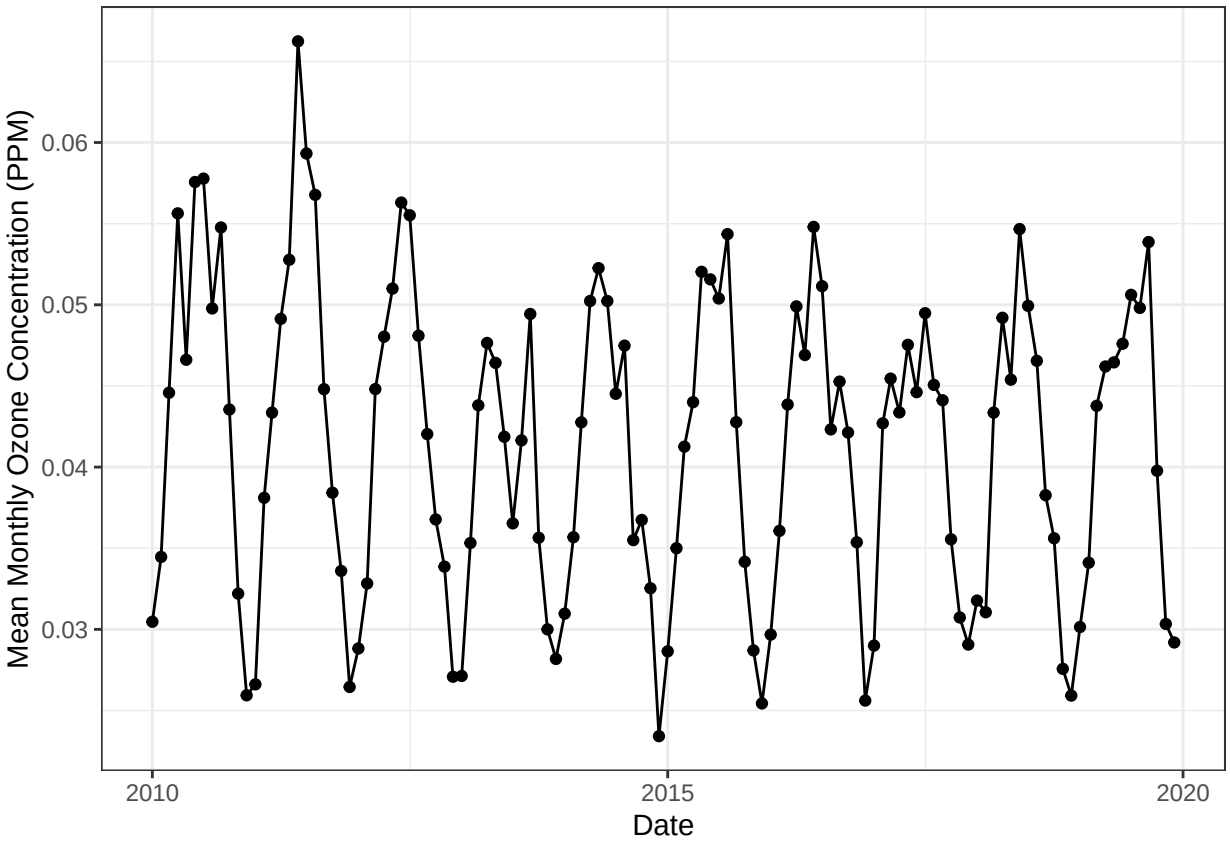
```
summary(Ozone.trend2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: OzoneGH.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
```

Answer: The seasonal Mann-Kendall is most appropriate, becuase of its clear seasonality with the Ozone dataset. Moreover, there is a cyclical up and down as the season change.

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
monthlyozoneplot<-
  ggplot(OzoneGH.monthly,
         aes(x= month_year, y= MeanMonthlyOzone))+
  geom_point()+
  geom_line()+
  ylab("Mean Monthly Ozone Concentration (PPM)")+
  xlab("Date")+
  theme_bw()

print(monthlyozoneplot)
```



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: From the graph, I can see the monthly ozone data started in 2010,they had significantly seasonal trend. I found that it reached the lowest in 2015, but got the highest data in 2011 to 2012.

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

GaringerOzoneNoSeasonal.ts<-
  OzoneGH.monthly.decompose$time.series[,2]+
  OzoneGH.monthly.decompose$time.series[,3]


#16
Ozone.trend3<-
  Kendall::MannKendall(GaringerOzoneNoSeasonal.ts)
Ozone.trend3
```

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

```
summary(Ozone.trend3)
```

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

```
Ozone.trend4<-
  trend::mk.test(GaringerOzoneNoSeasonal.ts)
Ozone.trend4
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzoneNoSeasonal.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: As the result shows, the p value is 0.0075, which is close to zero. It indicate that it has significant non seasonal ozone monthly series.