Assignment 8: Time Series Analysis

Yilin Zhong

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
getwd()
```

```
## [1] "C:/Users/victo/Desktop/Spring 2023/EDA/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## 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
```

```
library(Kendall)
```

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

```
library(tseries)
```

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

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

```
library(here)
```

```
## here() starts at C:/Users/victo/Desktop/Spring 2023/EDA/EDA-Spring2023
```

```
library(ggthemes)

my_theme<-theme_base()+
  theme(
     legend.background = element_rect(
      color='grey',
      fill = 'white'),
    plot.background = element_rect(
      color = 'white'),
     plot.title = element_text(
      color = 'lightblue'),
    legend.title = element_text(
      color = 'red')
  )

theme_set(my_theme)
```

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
ozone.2010<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                        stringsAsFactors = T)
ozone.2011<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                        stringsAsFactors = T)
ozone.2012<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                        stringsAsFactors = T)
ozone.2013<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                        stringsAsFactors = T)
ozone.2014<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                        stringsAsFactors = T)
ozone.2015<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                        stringsAsFactors = T)
ozone.2016<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                        stringsAsFactors = T)
ozone.2017<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                        stringsAsFactors = T)
ozone.2018<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                        stringsAsFactors = T)
ozone.2019<-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                        stringsAsFactors = T)

GaringerOzone<-rbind(ozone.2010,ozone.2011,ozone.2012,ozone.2013,ozone.2014,
                     ozone.2015,ozone.2016,ozone.2017,ozone.2018,ozone.2019)
dim(GaringerOzone)
```

```
## [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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

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

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

#5
Days<-as.data.frame(seq(as.Date("2010-01-01"),as.Date("2019-12-31"),by = 1))
colnames(Days)<-c("Date")
#6
GaringerOzone<-left_join(Days,GaringerOzone.wrangled)
```

```
## Joining, by = "Date"
```

```
dim(GaringerOzone)
```

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

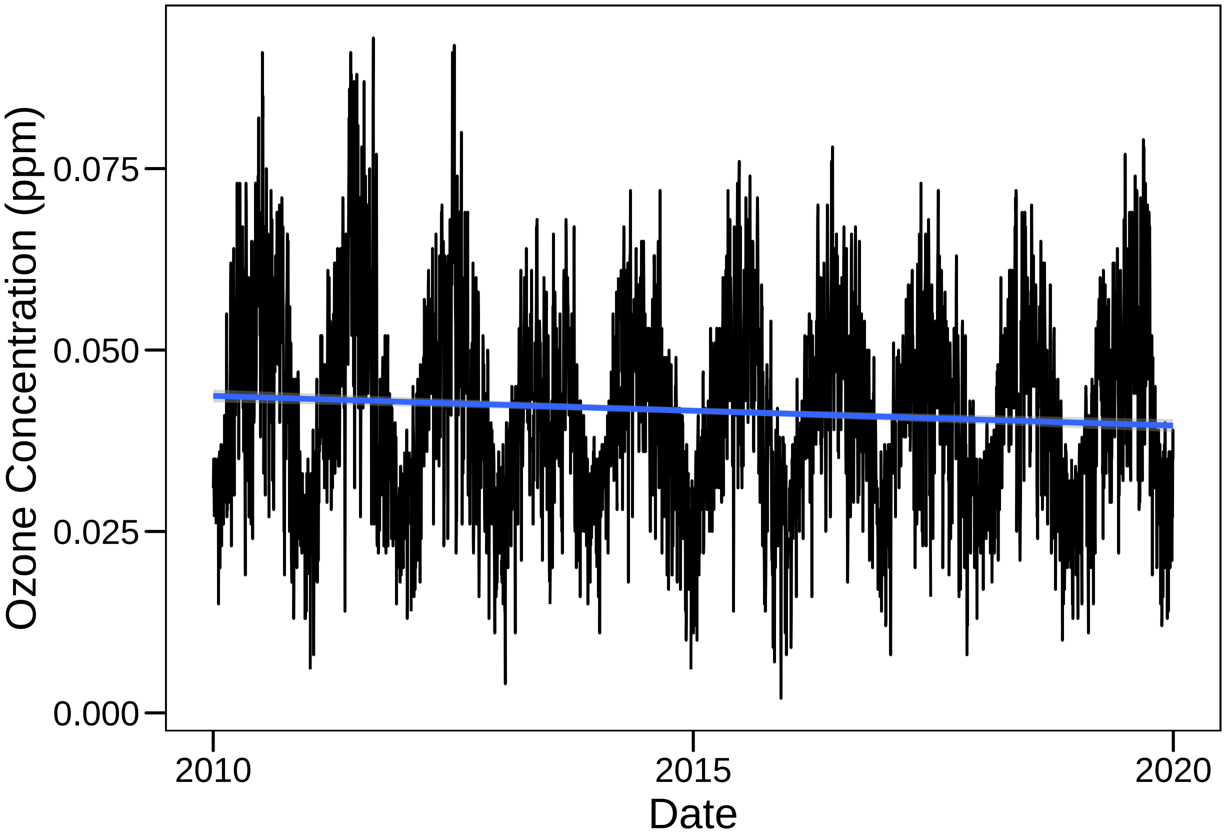
Visualize

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() +
  labs(x = "Date", y = "Ozone Concentration (ppm)")+
  geom_smooth( method = lm )
```

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

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



Answer: The plot does suggest a somewhat decreasing trend in ozne concentration over time as the smoothed line move downward overtime.

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
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
```

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

Answer: We didn't use a piecewise constant because in this approach any missing data are assumed to be equal to the measurement made nearest to that date. However, when we look at the plot we created in last question, there are actually variations in the concentration overtime, so making missing data equal to the measurement made nearest to that date may not be a good fit. As for spline interpolation, a quadratic function is used to interpolate rather than drawing a straight line. According to the plot in the last question, the variation is small, so a straight line is probably better than using a quadratic function.

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 %>%
  mutate(month = month(Date),
         year = year(Date)) %>%
  group_by(month, year) %>%
  summarize(monthly.mean.ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly$Date <- as.yearmon(paste(GaringerOzone.monthly$year,
                                               GaringerOzone.monthly$month), "%Y %m")
GaringerOzone.monthly$Date <-as.Date(GaringerOzone.monthly$Date,format = "%m %Y")
class(GaringerOzone.monthly$Date)
```

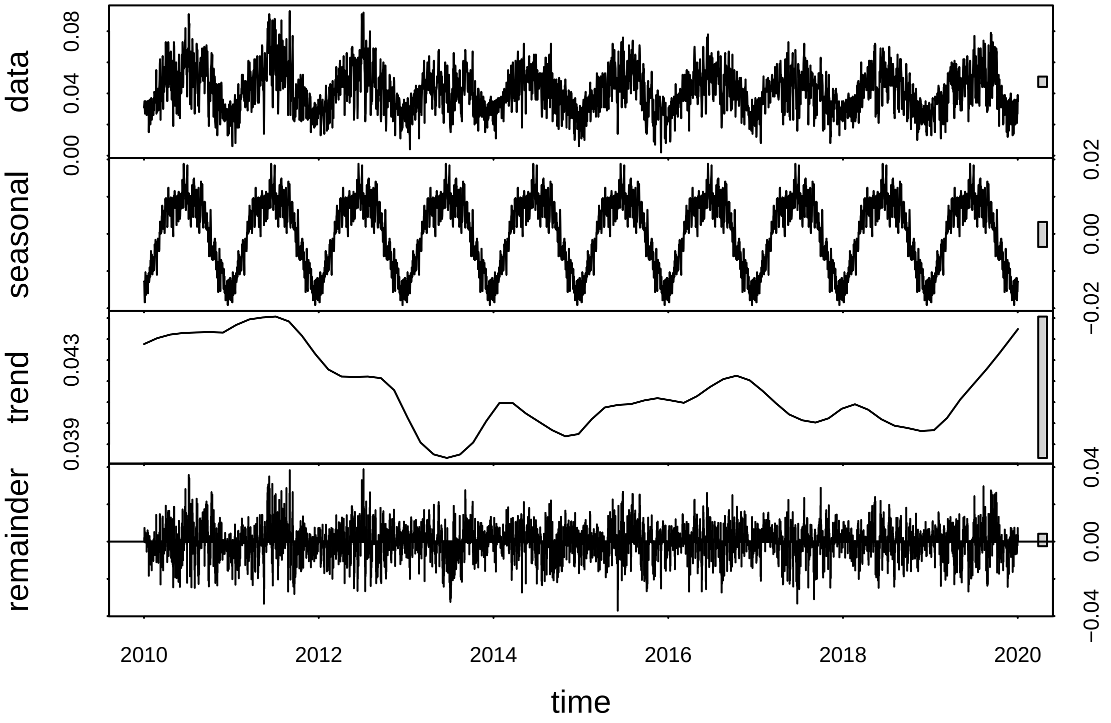
```
## [1] "Date"
```

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.

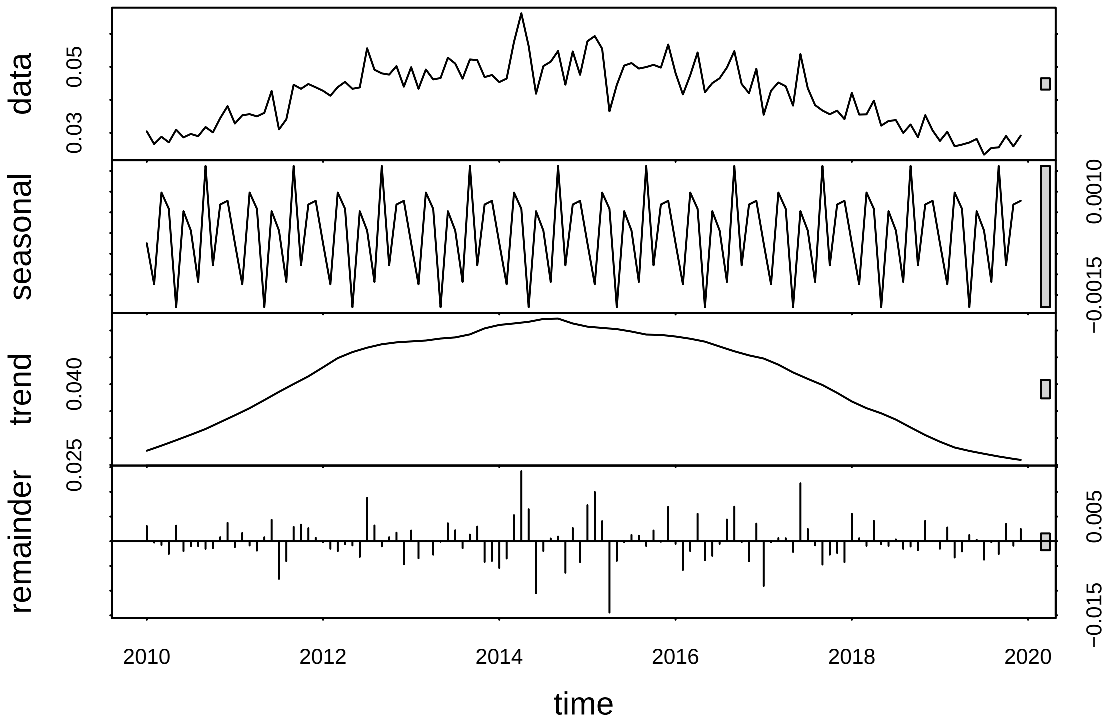
```
#10
f_month.daily <- month(first(GaringerOzone$Date))
f_year.daily <- year(first(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                   start=c(f_year.daily,f_month.daily),
                   frequency=365)

f_month.monthly <- month(first(GaringerOzone.monthly$Date))
f_year.monthly <- year(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$monthly.mean.ozone,
                   start=c(f_year.monthly,f_month.monthly),
                   frequency=12)
```

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

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



```
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?

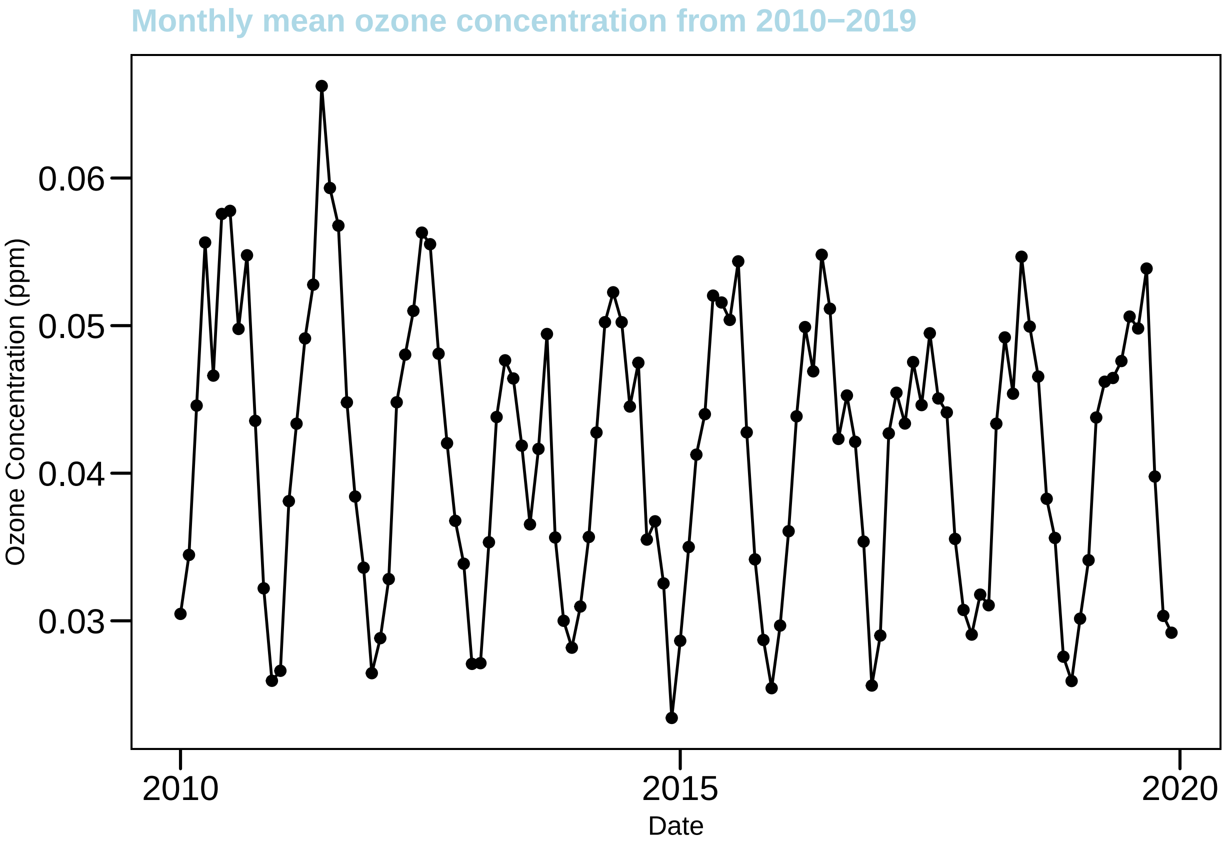
```
#12
GaringerOzone.monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly_trend)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: In this case the seasonal Mann-Kendall is most appropriate because according to the plot in above, monthly mean ozone concentration has a seasonal cycle. Thus, we are interested in in knowing how monthly mean ozone concentration has changed over the course of measurement while incorporating the seasonal component. The use of Seasonal Mann-Kendall test allow us to figure out whether a monotonic trend exists.

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
ozone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y =monthly.mean.ozone )) +
  geom_point() +
  geom_line() +
  labs(y= "Ozone Concentration (ppm)",
       title = "Monthly mean ozone concentration from 2010-2019")+
  theme(title = element_text(size = 10))

print(ozone.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: According to the grapgh that we created, there is a seaonal cycle for ozone concentration, which we use the Seasonal Mann-Kendall test to figure out whether a monotonic trend exists. The null hypothesis for Seasonal Mann-Kendall test is that the there is no trend present in the data. The result of the test show a score of -54 that suggest a deccreasing trend exists. However, the test has a p-value of 0.163, which means the trend is not statiscally signifcant and there is not enough evidence to reject the null hypothesis (p-value=0.163). Thus, the test reveals that there is no trend presend in the data, which means ozone concentrations has not changed over the 2010s at this station.

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.

```r
#15
GaringerOzone.monthly_Components <- as.data.frame(GaringerOzone.monthly_decomp$time.series[,1:3])
GaringerOzone.monthly_Components <- mutate(GaringerOzone.monthly_Components,
        Observed = GaringerOzone.monthly$monthly.mean.ozone,
        Date = GaringerOzone.monthly$Date)

#16
GaringerOzone.nonseasonal.monthly.ts <- ts(GaringerOzone.monthly_Components$remainder,
                   start=c(f_year.monthly,f_month.monthly),
                   frequency=12)
GaringerOzone.nonseasonal.monthly_trend <- Kendall::MannKendall(GaringerOzone.nonseasonal.monthly.ts)
summary(GaringerOzone.nonseasonal.monthly_trend)
```

```
## Score =  -118 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.0165, 2-sided pvalue =0.79071
```

> Answer: The result on the non-seasonal Ozone monthly series is the same with the results on the ones obtained with the Seasonal Mann Kendall on the complete series as the p-value for the Mann Kendall test is 0.791, which it does not have enough evidence to rejects the null hypothesis. Thus, there is no trend present in the data, which the ozone concentrations has not changed over the 2010s at this station.