Assignment 7: Time Series Analysis

Yosia Theo Napitupulu

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot 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.

```
#1 Working directory
getwd()
```

```
## [1] "E:/ENV872/EDA-Fall2022"
```

```
library(tidyverse)
library(lubridate)
#install.packages("zoo")
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.2.2
```

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

```
## Warning: package 'trend' was built under R version 4.2.2
```

```
#install.packages("tseries")
library(Kendall)
#library(tseries)
library(ggplot2)
library(dplyr)


# 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
GaringerOzone10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
GaringerOzone11 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
GaringerOzone12 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
GaringerOzone13 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
GaringerOzone14 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
GaringerOzone15 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
GaringerOzone16 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
GaringerOzone17 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
GaringerOzone18 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
GaringerOzone19 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone.main <- rbind(GaringerOzone10, GaringerOzone11, GaringerOzone12, GaringerOzone13, Garinge
class (GaringerOzone.main)
```

```
## [1] "data.frame"
```

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

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

```
# 4 Select new dataset
GaringerOzone.data <- GaringerOzone.main %>% select("Date", Daily_O3_Concentration = "Daily.Max.8.hour.
head(GaringerOzone.data)
```

```
##         Date Daily_O3_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 2010-01-07                 0.033              31
```

```
# 5 Add new dataframe in daily basis
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = 1), colnar
colnames(Days) <- "Date"
head(Days)
```

```
##         Date
## 1 2010-01-01
## 2 2010-01-02
## 3 2010-01-03
## 4 2010-01-04
## 5 2010-01-05
## 6 2010-01-06
```

```
# 6 Combine again the dataframe
GaringerOzone <- left_join(Days, GaringerOzone.data, by = c("Date"))
head(GaringerOzone)
```

```
##         Date Daily_O3_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
```

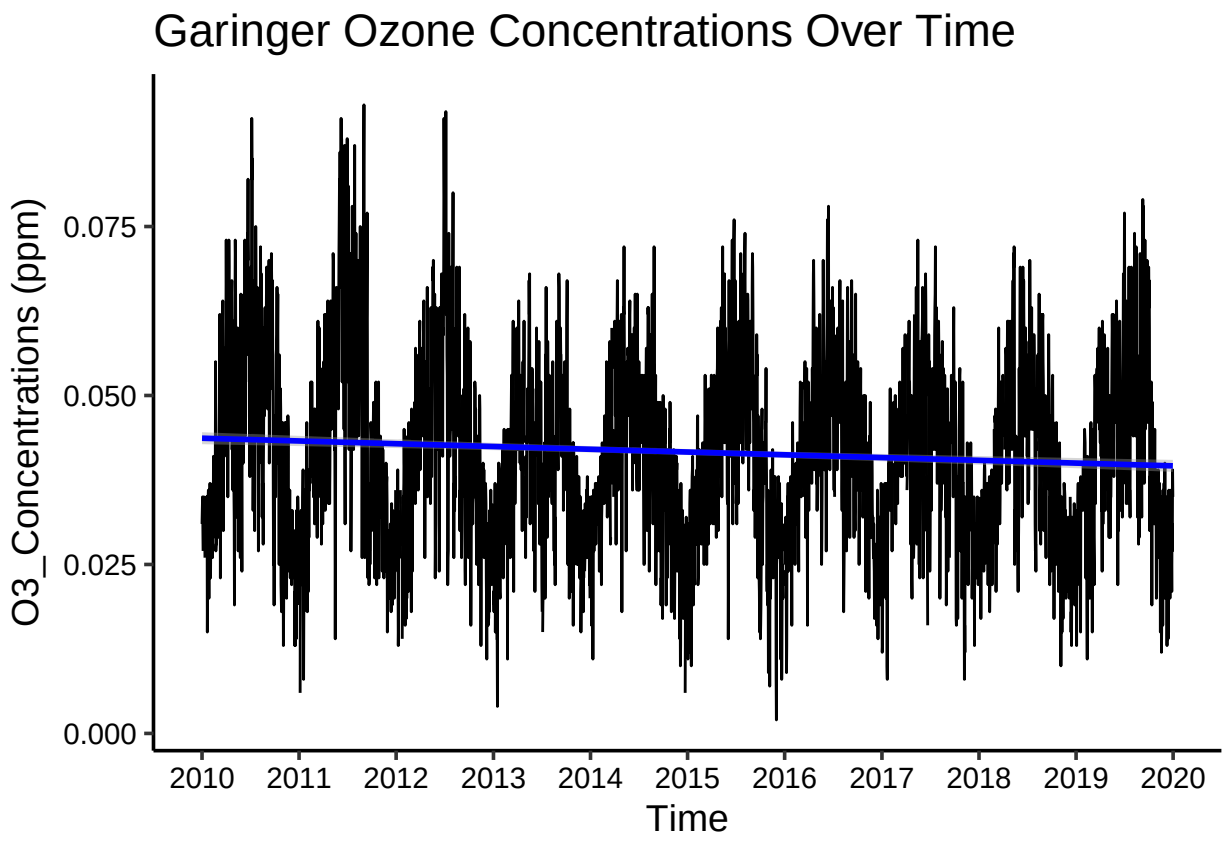
Visualize

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 Create a line plot depicting ozone concentrations over time
ggplot(GaringerOzone, aes(y = Daily_O3_Concentration , x = Date),
  size = 0.25) + geom_line() +
  geom_smooth(method = "lm", color = "blue") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  ylab(expression("O3_Concentrations (ppm)")) +
  xlab(expression("Time")) +
  ggtitle("Garinger Ozone Concentrations Over Time")
```

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

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



Answer: Yes, it does have a decreasing trend over time, with a sloping line as shown in the graph.

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 linear interpolation to fill missing daily data for Ozone concentration
summary(GaringerOzone)
```

```
##       Date            Daily_O3_Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200        Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200        1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100        Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163        Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100        3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300        Max.   :169.00
##                       NA's   :63             NA's   :63
```

```
#GaringerOzone.linear <- GaringerOzone %>% mutate(Daily_O3_Concentration = zoo::na.approx(Daily_O3_Conc

GaringerOzone$Daily_O3_Concentration <- na.approx(GaringerOzone$Daily_O3_Concentration)
GaringerOzone$DAILY_AQI_VALUE <- na.approx(GaringerOzone$DAILY_AQI_VALUE)

#Note the NA is gone
summary(GaringerOzone)
```

```
##       Date            Daily_O3_Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200        Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200        1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100        Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151        Mean   : 41.41
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100        3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300        Max.   :169.00
```

Answer: We do not use piecewise constant because in filling the missing data with the equal to the nearest value would not suit particularly when drawing a straight line. Spline interpolation uses a quadratic function to interpolate rather than drawing a straight line seems too complex. Meanwhile, linear interpolation has linear value which more simple and feasible in connecting the dots to drawing the 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 Generating aggregated data mean ozone concentrations for each month
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(month_year = floor_date(Date, "month")) %>%
  group_by(month_year) %>%
  summarise(mean.concentration = mean(Daily_O3_Concentration))
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 2
##   month_year mean.concentration
##   <date>                  <dbl>
## 1 2010-01-01             0.0305
## 2 2010-02-01             0.0345
```

```
## 3 2010-03-01             0.0446
## 4 2010-04-01             0.0556
## 5 2010-05-01             0.0466
## 6 2010-06-01             0.0576
```

```
#GaringerOzone.monthly <- GaringerOzone %>%  mutate(Month = month(Date), Year = year(Date)) %>% mutate(
```

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.

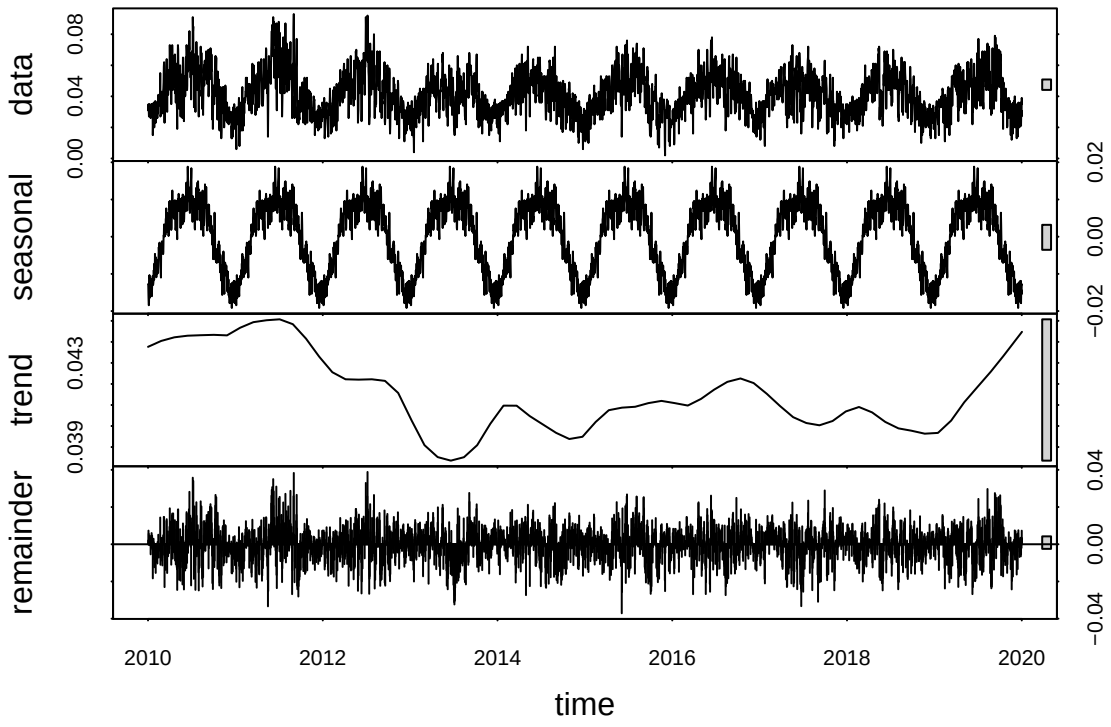
```
#10 Generate two time series objects daily
GaringerOzone.daily.ts <-
  ts(GaringerOzone$Daily_O3_Concentration, start = c(2010,01,01), frequency=365)

# Generate two time series objects monthly
GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$mean.concentration, start=c(2010,01), frequency=12)
```

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

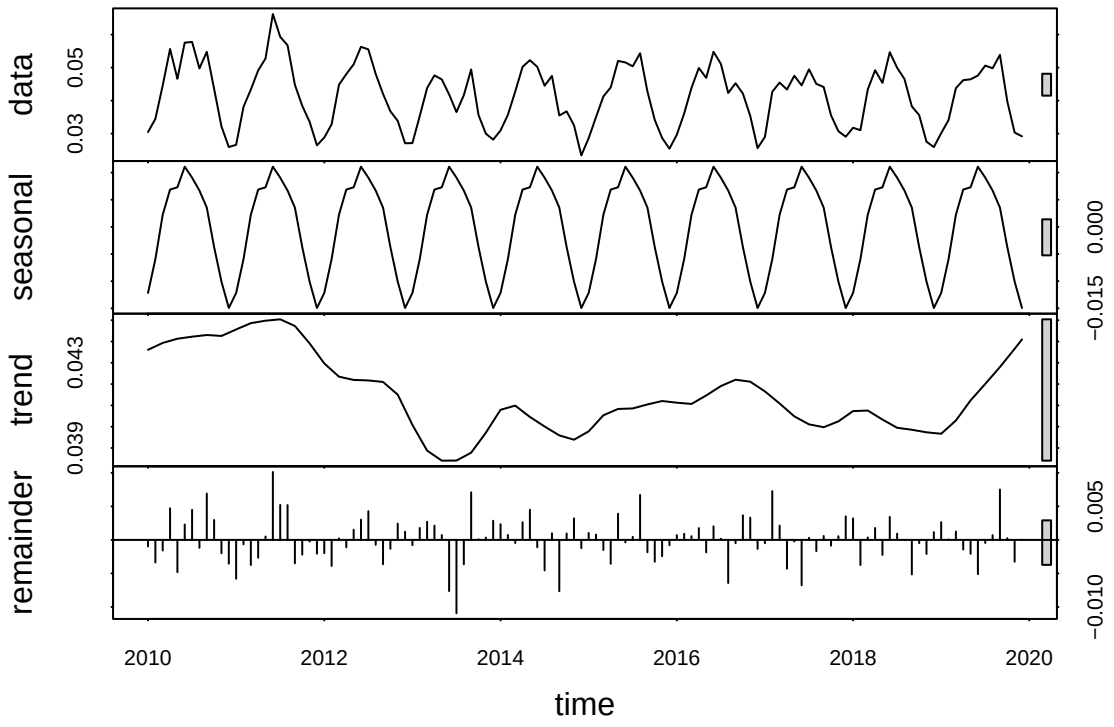
```
#11 Decompose the daily time series
GaringerOzone.daily.ts_decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.ts_decomp)
```




```
#Decompose the monthly time series
GaringerOzone.monthly.ts_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(GaringerOzone.monthly.ts_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 Run monotonic trend analysis using seasonal Mann-Kendall
SMK_test <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
print(SMK_test)
```

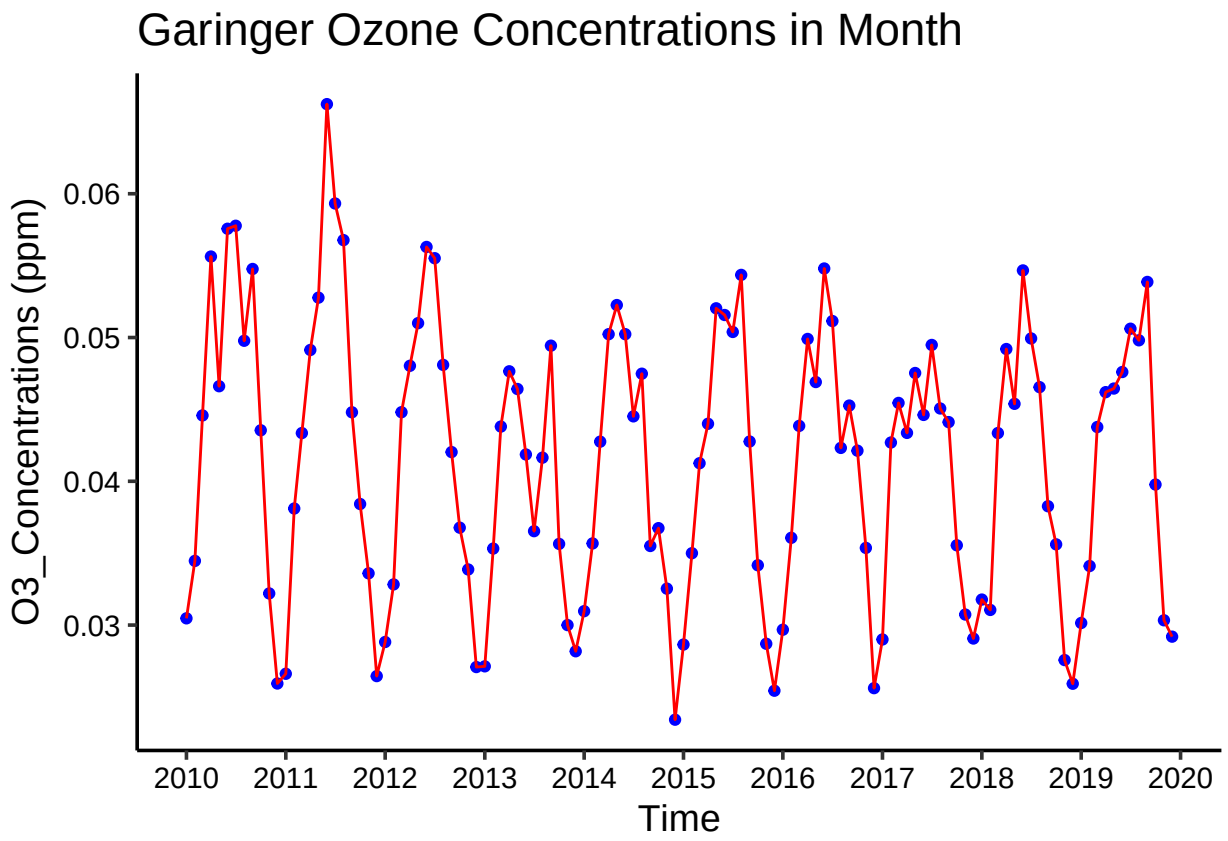
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall become the most appropriate trend analysis because the series shows a clear seasonal pattern over the observation period.

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 Plot depicting mean monthly ozone concentrations over time
ggplot(GaringerOzone.monthly, aes(x = month_year, y = mean.concentration)) +
  geom_point(color = "blue") +
  geom_line(color = "red", method = "lm") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  xlab("Time") +
  ylab(expression("O3_Concentrations (ppm)")) +
  ggtitle("Garinger Ozone Concentrations in Month")
```

```
## Warning: Ignoring unknown parameters: method
```



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.

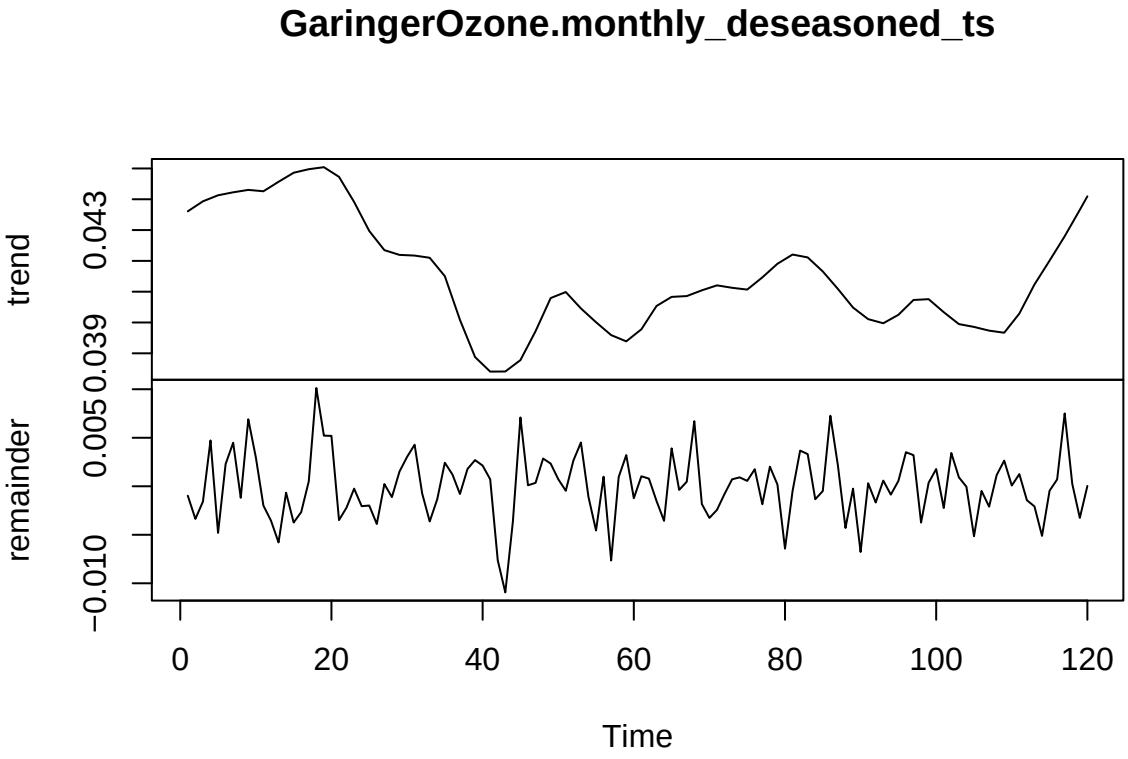
    Answer: According to the monthly average of the Ozone concentration, the graph shows a steady increase.The p-value from the Seasonal Mann Kendall shows that we can reject the null hypotesis Ho.

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 Subtract the seasonal component
GaringerOzone.monthly_deseasoned <- as.data.frame(GaringerOzone.monthly.ts_decomp$time.series[,2:3])


#16 Run the Mann Kendall test on the non-seasonal Ozone monthly series
GaringerOzone.monthly_deseasoned_ts <- ts(GaringerOzone.monthly_deseasoned)
plot(GaringerOzone.monthly_deseasoned_ts)
```




```
MK_test <- MannKendall(GaringerOzone.monthly_deseasoned_ts)
print(MK_test)
```

```
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

Answer: For the Mann Kendall test, the p-value is greater than the significance level, then we reject the null hypothesis.As seasonal Mann Kendall test gives the overall trend in series by looking at the each month trend.