Assignment 7: Time Series Analysis

Shidi Dai

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.

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=80), tidy=TRUE)
#1
getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```
library(tidyverse)
#install.packages(lubridate)
library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
options(scipen = 4)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top",
        legend.title = element_text(size=10),
```

```
        legend.text = element_text(size=7))
theme_set(mytheme)

#2
GaringerNC2010 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = TRUE)
GaringerNC2011 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = TRUE)
GaringerNC2012 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = TRUE)
GaringerNC2013 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = TRUE)
GaringerNC2014 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = TRUE)
GaringerNC2015 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = TRUE)
GaringerNC2016 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = TRUE)
GaringerNC2017 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = TRUE)
GaringerNC2018 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = TRUE)
GaringerNC2019 <- read.csv(
  "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = TRUE)

GaringerOzone <- rbind(
  GaringerNC2010, GaringerNC2011, GaringerNC2012, GaringerNC2013, GaringerNC2014,
  GaringerNC2015, GaringerNC2016, GaringerNC2017, GaringerNC2018, GaringerNC2019)
```

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")

# 4
GaringerOzone_Processed <- GaringerOzone %>%
    select(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 day"))
colnames(Days)[1] <- "Date"
```

```
# 6
GaringerOzone <- left_join(Days, GaringerOzone_Processed)
```

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

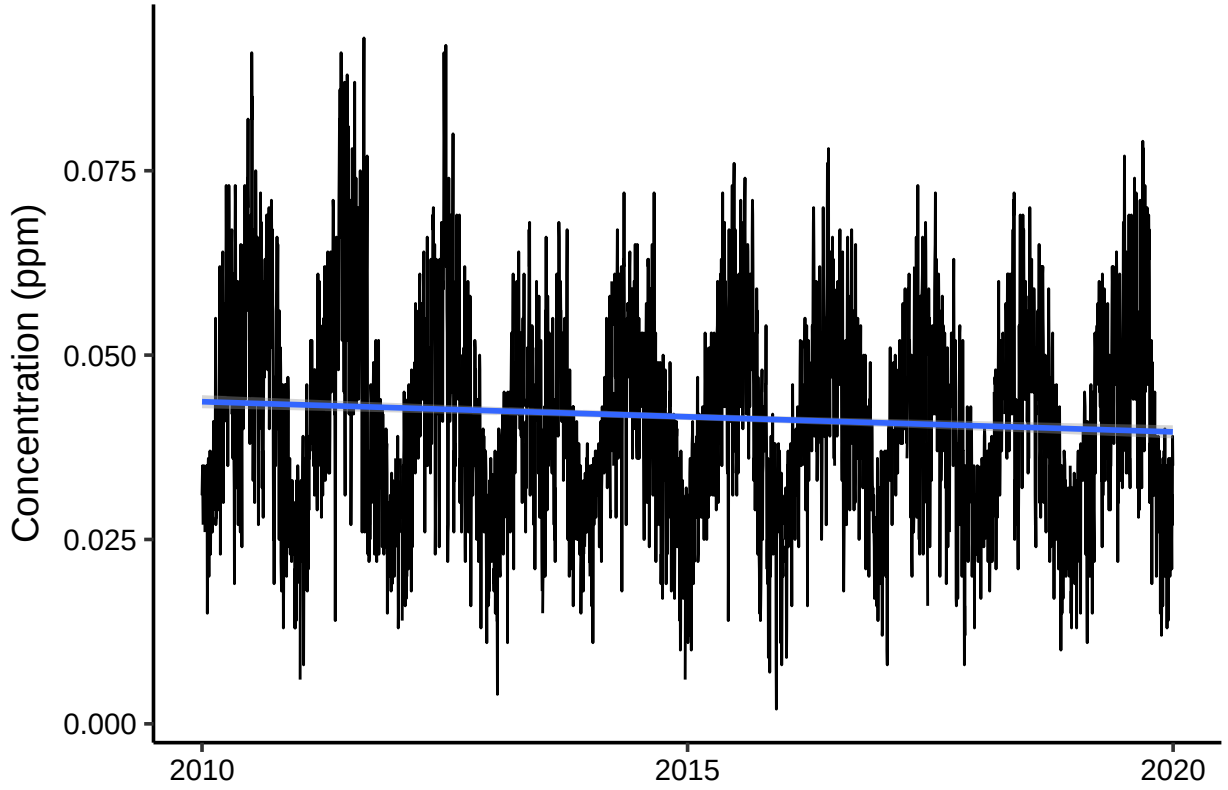
Visualize

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

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

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



Answer: The smoothed line shows a slightly downward linear trend of the data which indicates a slightly negative relationship between concentration and date. There is an overall decrease tendency over the relationship. As date increases, concentration decreases.

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 <- GaringerOzone %>%
    mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Piecewise constant will make the missing data equal to the value of the one at the nearest date (late or early). From the plot we can clearly see there is a trend between dates and concentration. So use the previous or late data for estimating the NAs are not accurate. Spline interpolation uses a quadratic function instead of a linear function to interpolate. There is no need here to use a quadratic function as we see some linear trend from our plot.

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)

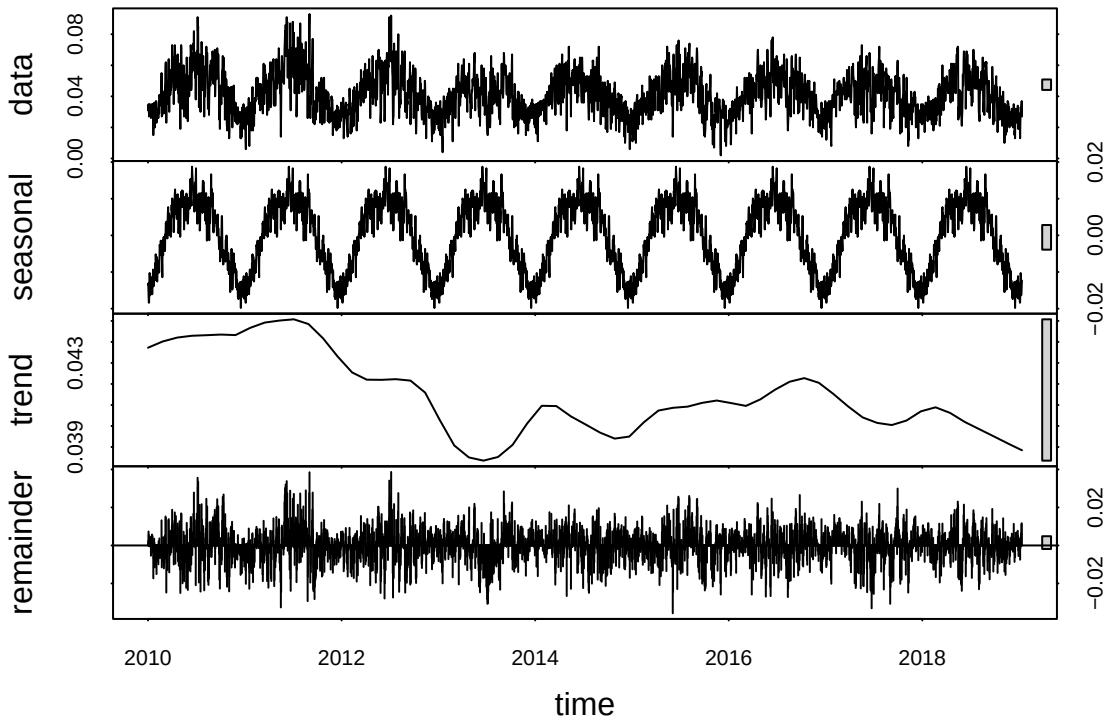
```
# 9
GaringerOzone_1 <- GaringerOzone %>%
    mutate(Month = month(GaringerOzone$Date)) %>%
    mutate(Year = year(GaringerOzone$Date))
GaringerOzone.monthly <- aggregate(Daily.Max.8.hour.Ozone.Concentration ~ Month +
    Year, GaringerOzone_1, FUN = mean) %>%
    mutate(Date = seq(as.Date("2010-01-01"), as.Date("2019-12-01"), by = "1 month"))
colnames(GaringerOzone.monthly)[3] <- "Mean.Ozone.Concentration"
GaringerOzone.monthly <- GaringerOzone.monthly %>%
    select(Date, Mean.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.

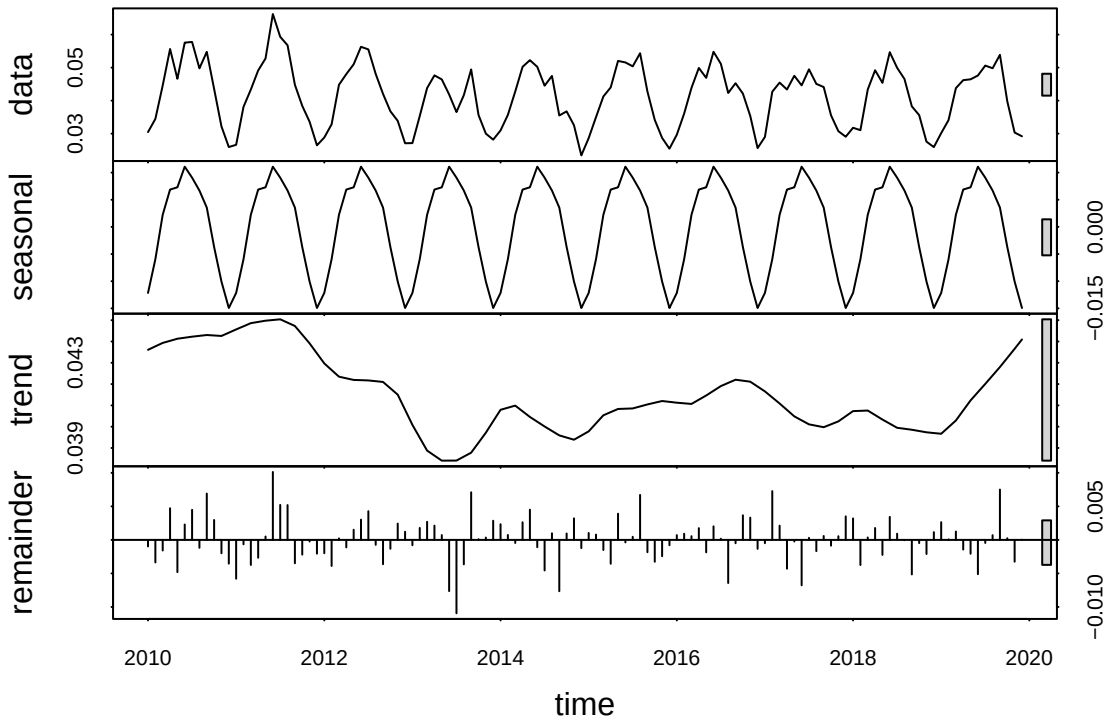
```
# 10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
    start = c(2010, 1), end = c(2019, 12), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Ozone.Concentration, start = c(2010,
    1), end = c(2019, 12), frequency = 12)
```

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

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




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



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_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone_trend1
```

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

```
summary(GaringerOzone_trend1)
```

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

```
GaringerOzone_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
GaringerOzone_trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.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(GaringerOzone_trend2)
```

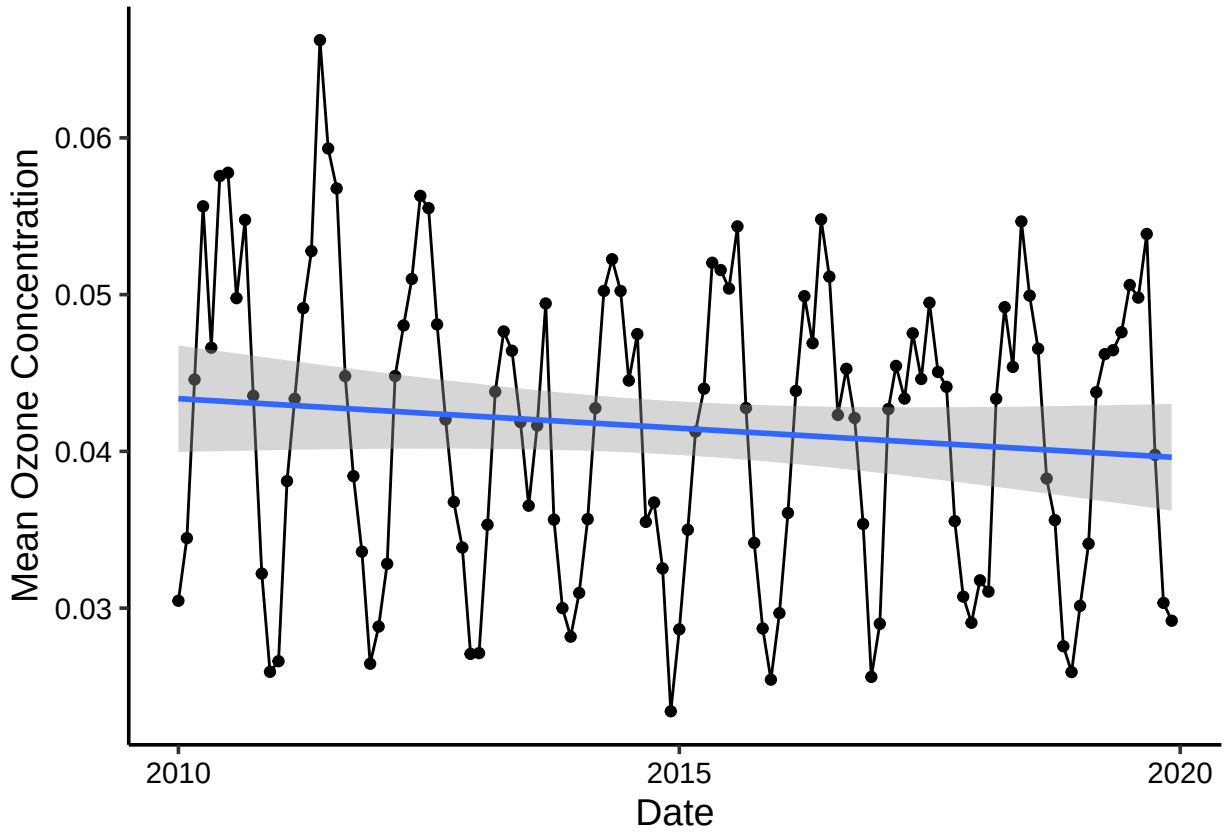
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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 because the trend is not linear but there is seasonality (which repeats over a fixed known period). The seasonal Mann-Kendall is made to handle non-linear data set with seasonality.This is also the only option to use when we have 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.

```
# 13
GaringerOzone_plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Ozone.Concentration)) +
    geom_point() + geom_line() + ylab("Mean Ozone Concentration") + geom_smooth(method = lm)
print(GaringerOzone_plot)
```

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



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: We are going to reject the null hypothesis (which says that the data is stationary). We conclude that there is a trend between monthly mean ozone concentration and date (p=0.046724<0.05). From the smk test, we can see the trend of each season (month). We can see that for some season, there is a stronger tendency of decrease (season 4 and 6 are -17, and season 5 is -15). For some seasons, there is less or no tendency of decrease (season 1 is 15 and season 12 is 11). But overall, the p value is 0.04965<0.05 so we reject the null hypothesis which said that the trend is stationary (S=0). Thus, there is a negative trend on monthly mean ozone concentration overtime.

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
GaringerOzone.monthly_Components <- GaringerOzone.monthly_decomposed$time.series[,
    2:3]

# 16
GaringerOzone_trend3 <- Kendall::MannKendall(GaringerOzone.monthly_Components)
GaringerOzone_trend3
```

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

```
summary(GaringerOzone_trend3)
```

```
## Score =  -16300 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

Answer: The results from #16 also shows that we reject the null hypothesis (p =< 2.22e-16 < 0.05). So we reject the hypthosis which said that the trend is stationay. Thus, we conclude that there is a trend in mean Ozone concentration over month. This result matches what we got from using the Seasonal Mann Kendall.