Assignment 7: Time Series Analysis

Israel Golden

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 Monday, March 14 at 7:00 pm.

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] "/Users/israelgolden/Desktop/School/MEM/Semester 4/ENV 872/GitHub Repos/Environmental_Data_Analy
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- 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)
library(Kendall)

#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 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
# packages
require(data.table)

# set wd
setwd("../Data/Raw/Ozone_TimeSeries/")

# import files
ozonetimeseries_files = list.files(pattern="*.csv")
dataset = do.call(rbind, lapply(ozonetimeseries_files, fread))
rm(ozonetimeseries_files)
# transform data to df
GaringerOzone <- as.data.frame(unclass(dataset))

setwd("/Users/israelgolden/Desktop/School/MEM/Semester 4/ENV 872/GitHub Repos/Environmental_Data_Analyti
setwd("./")
```

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.

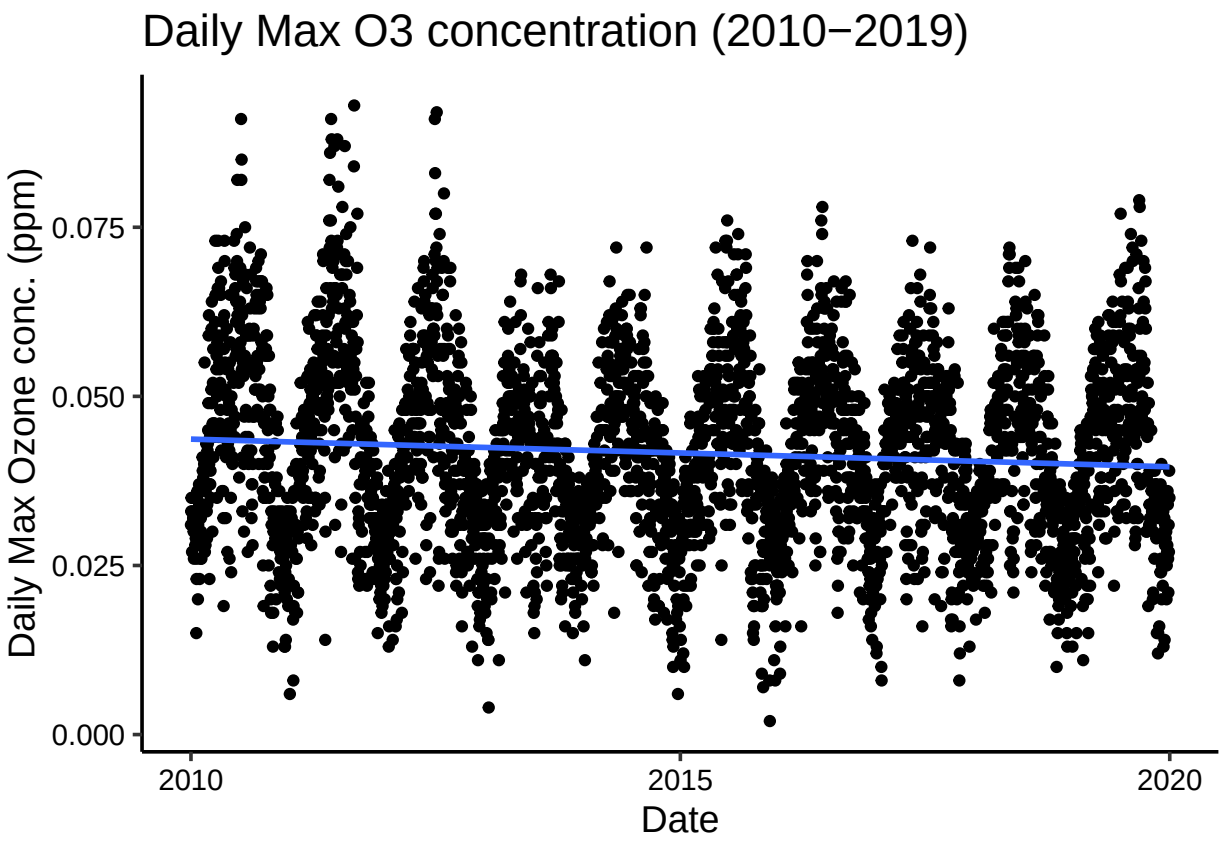
```
# 3
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

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

# 5
start_date <- as.Date("2010/01/01")
end_date <- as.Date("2019/12/31")
Days <- as.data.frame(seq(start_date, by = "day", end_date))

colnames(Days)[colnames(Days) == "seq(start_date, by = \"day\", end_date)"] <- "Date"

# 6
GaringerOzone <- left_join(Days,GaringerOzone,"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_point() +
  labs(x = "Date", y = "Daily Max Ozone conc. (ppm)", title = "Daily Max O3 concentration (2010-2019)")
  geom_smooth(method = lm, se = FALSE)
```

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

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

## Warning: Removed 63 rows containing missing values (geom_point).
```



Answer: While it is evident there is some seasonality in the data, the overall trend appears to be sloping gently downward. The seasonality in the data (i.e., higher levels of Ozone during Summer months) is likely due to increased heat and use of combustion engines (i.e., cars) during this season.

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
```

```
# We have 63 NA's in our dataframe for Daily Max 8-hour Ozone concentration

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

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
# Now we have 0 NAs!
```

Answer: In the linear interpolation approach, missing data are interpolated with values that fall between the previous and subsequent measurement. This method assumes that there interstitial data do not significantly deviate from bordering data. Spline is similar to the linear interpolation approach except that a quadratic formula is used to interpolate points rather than a linear formula. We did not use spline because the data can be fit reasonably well with a linear approach and it requires fewer assumptions about the data. Piece-wise constant interpolation approach assumes that the N/A values are equivalent to the nearest known data point. Based on the observed trends in the data - specifically its seasonality - it is probably unreasonable to assume that missing data will be equivalent to its nearest neighbor, rather than some midway point between previous and subsequent data points.

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)) %>%
  mutate(year = year(Date)) %>%
  group_by(year, month) %>%
  summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(Date = my(paste0(month,"-",year))) %>%
  select(Date,mean_Ozone)
```

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

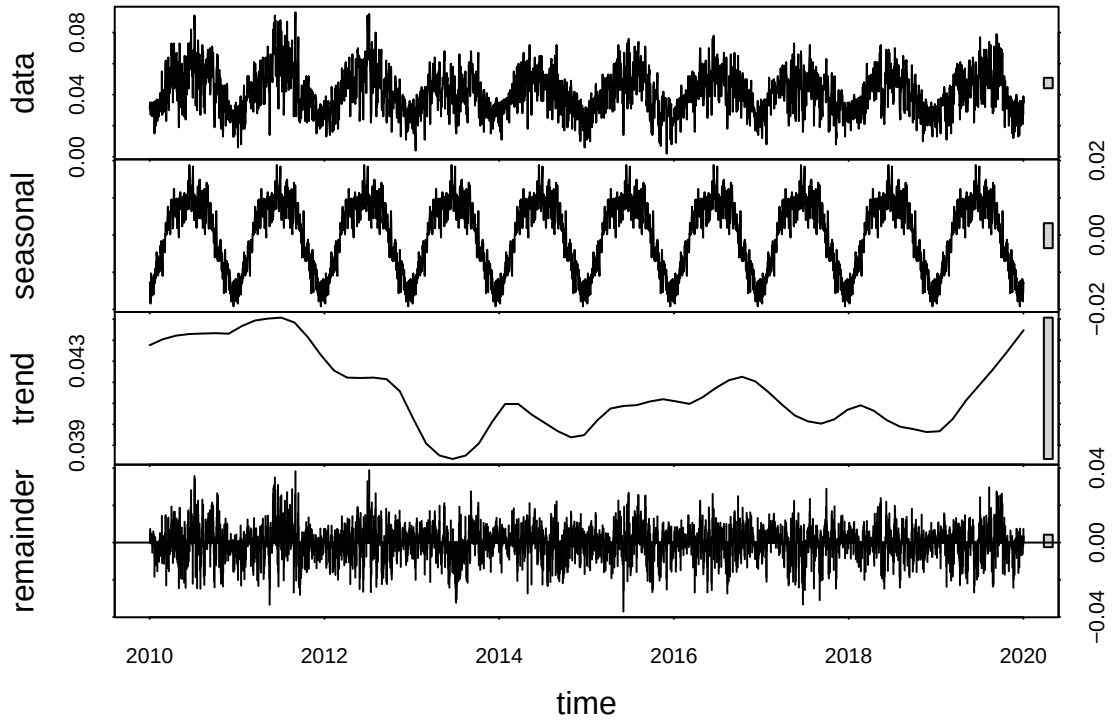
```
## Adding missing grouping variables: `year`
```

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.

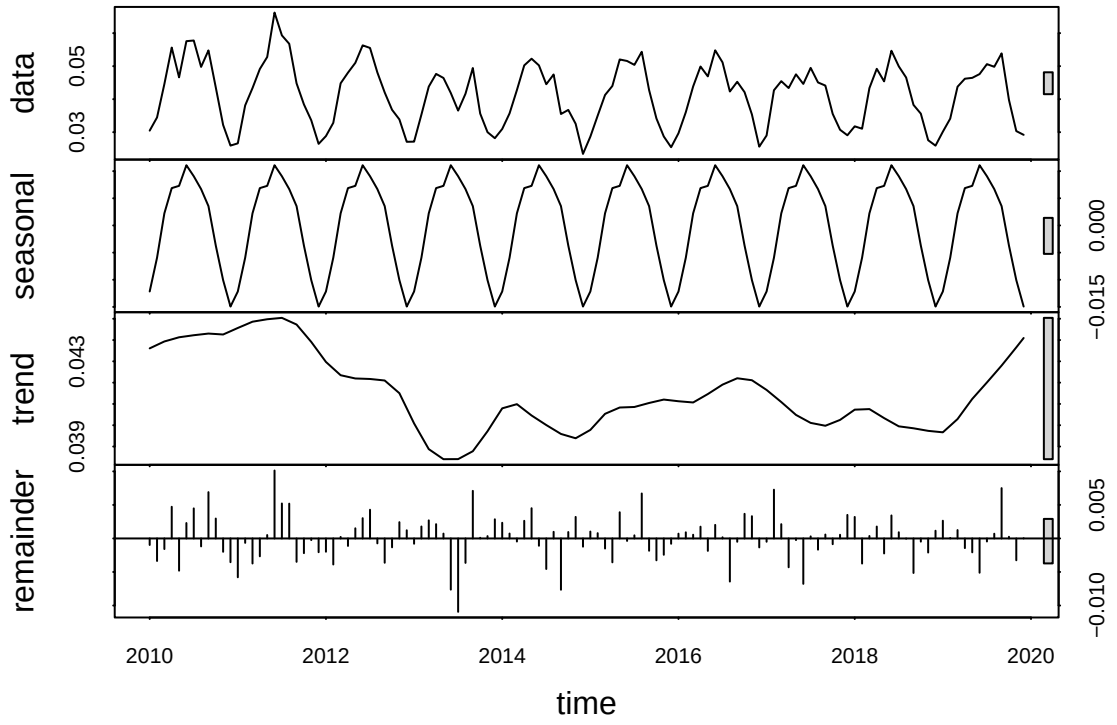
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,01,01),
                             frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_Ozone,
                               start = c(2010,01,01),
                               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.SMK <- trend::smk.test(GaringerOzone.monthly.ts)
GaringerOzone.monthly.SMK
```

```
##
##  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
```

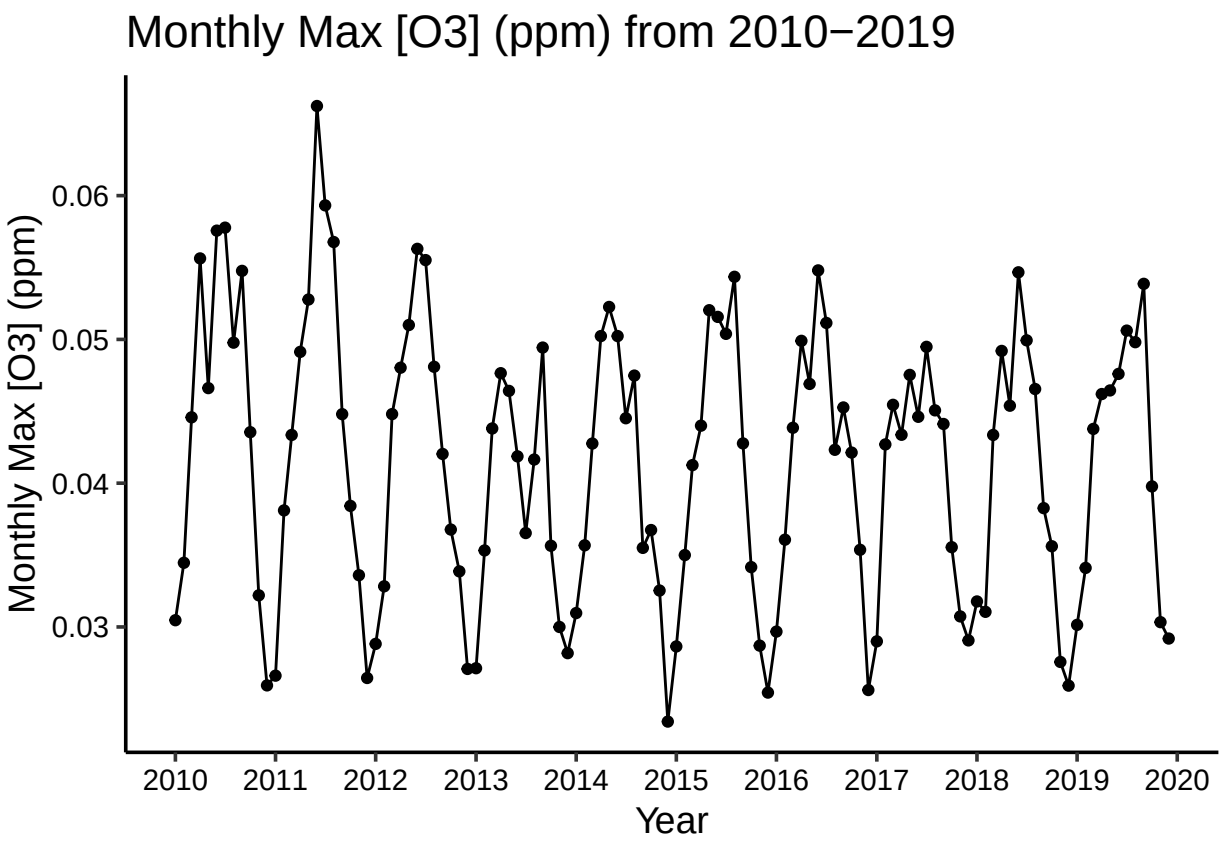
```
summary(GaringerOzone.monthly.SMK)
```

```
##
##  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: Seasonal Mann-Kendall is the most appropriate test for a monotonic trend analysis of this time series because there is a seasonal fluctuation in ozone concentration.

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 = Date,y = mean_Ozone)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_breaks = "years", date_labels = "%Y") +
  labs(x = "Year",
       y = "Monthly Max [O3] (ppm)",
       title = "Monthly Max [O3] (ppm) from 2010-2019")
```



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: In this study, we sought to determine if there existed seasonal trends in monthly maximum ozone concentration over a ten-year period Jan 2010 - Dec 2019) at Garinger High School in North Carolina. The null hypothesis, that no such monotonic trends existed in our dataset, was tested with the seasonal Mann-Kendall test. The results of the test were marginally significant ($p = 0.04$) such that we could reject the null in favor of the alternative hypothesis, that there are seasonal, monontonic trends in monthly maximum ozone concentration between 2010 and 2019. From the accompanying graph, it is evident that ozone concentrations have a peak in concentration in mid-summer and a nadir in mid-winter each year.

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.Components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

GaringerOzone.Components <- mutate(GaringerOzone.Components,
                                   Date = GaringerOzone.monthly$Date)
GaringerOzone.monthly.noseason.ts <- ts(GaringerOzone.Components$trend,
                                        start = c(2010,01,01), frequency = 12)

#16
GaringerOzone.monthly.MK <- MannKendall(GaringerOzone.monthly.noseason.ts)
GaringerOzone.monthly.MK
```

```
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

```
summary(GaringerOzone.monthly.MK)
```

```
## Score =  -1922 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

Answer: The p-value of the Mann Kendall test on the non-seasonal Ozone monthly series is significant (p<1.4e-05). This means we can reject the null hypothesis that there is not a monotonic upward or downward trend to the data in favor of our alternative hypothesis: that there is a trend in the data. This result has a significantly smaller p-value (by 5 orders of magnitude) than the seasonal trend. Based on the first plot produced for question 7, it appears that the overall trend is a gently sloping downward one.