Assignment 8: Time Series Analysis

Amanda Sajewski

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/asaje/EDA-Spring2023/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

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

```r
library(zoo)
```

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

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(trend)
```

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

```r
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
mytheme<- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "side")
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
ozone1<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2010_raw.csv",
                  stringsAsFactors = T)
ozone2<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2011_raw.csv",
                  stringsAsFactors = T)
ozone3<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2012_raw.csv",
                  stringsAsFactors = T)
ozone4<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2013_raw.csv",
                  stringsAsFactors = T)
ozone5<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2014_raw.csv",
                  stringsAsFactors = T)
ozone6<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2015_raw.csv",
                  stringsAsFactors = T)
ozone7<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2016_raw.csv",
                  stringsAsFactors = T)
ozone8<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2017_raw.csv",
                  stringsAsFactors = T)
ozone9<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2018_raw.csv",
                  stringsAsFactors = T)
ozone10<- read.csv("./Data/Raw/Ozone_Timeseries/EPAair_O3_GaringerNC2019_raw.csv",
                   stringsAsFactors = T)

GaringerOzone <- rbind(ozone1,ozone2,ozone3,ozone4,ozone5,ozone6,ozone7,ozone8,
                       ozone9,ozone10)
```

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
GaringerOzone1<- GaringerOzone %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

#5
Days<-
```

```r
  as.data.frame(seq(as.Date("2010/01/01"),as.Date("2019/12/31"),"days"))
colnames(Days)[1]= "Date"
class(Days$Date)
```

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

```r
class(GaringerOzone1$Date)
```

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

```r
#6
GaringerOzone <- left_join(Days,GaringerOzone1,by = "Date")

#For some reason this left_join returned all NA values until I selected which
#columns I wanted in my new data frame using a pipe function
```

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?

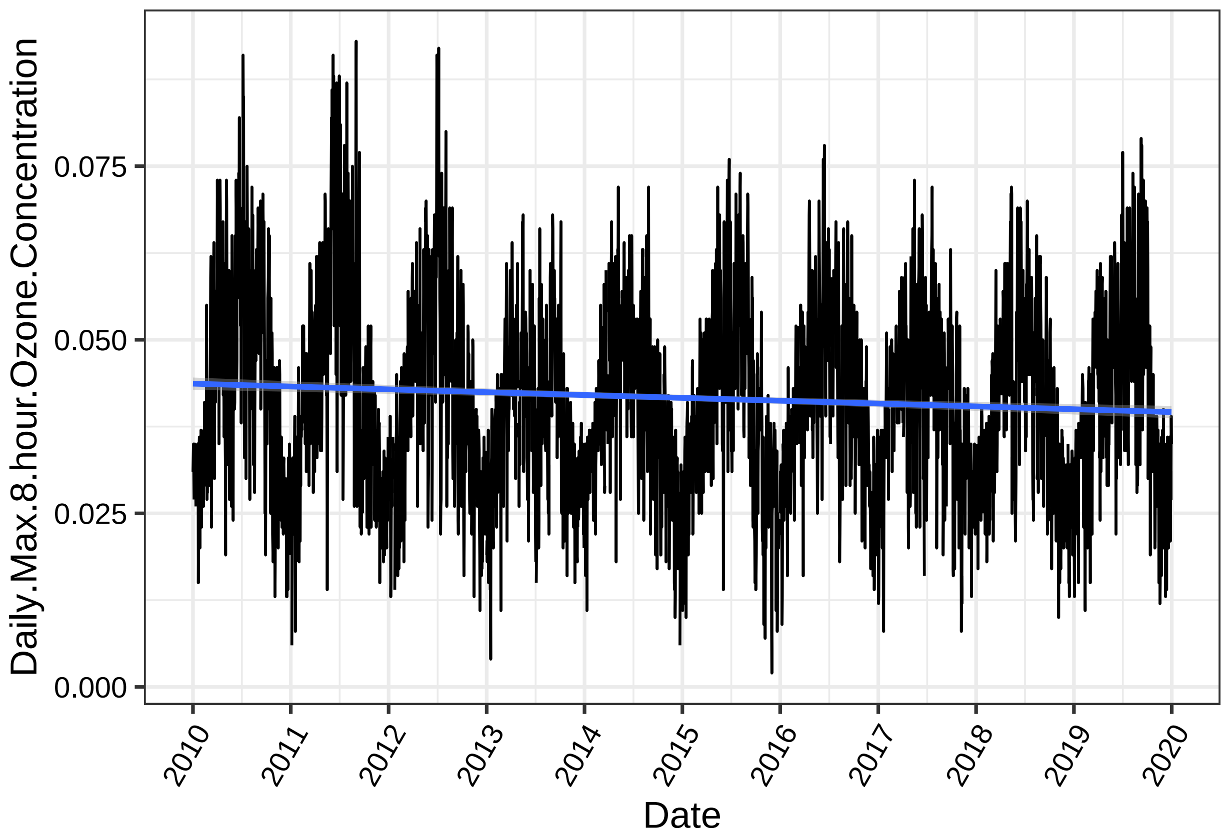
```r
#7

Ozoneconc<- ggplot(GaringerOzone, aes(x=Date,
                                      y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method=lm)+
  scale_x_date(labels = date_format("%Y"),
               date_breaks = "1 year")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
  labs( title = "Ozone over time",
        y="Ozone Concentrations (ppm)")
```

```
## $y
## [1] "Ozone Concentrations (ppm)"
##
## $title
## [1] "Ozone over time"
##
## attr(,"class")
## [1] "labels"
```

```r
print(Ozoneconc)
```

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

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



Answer: This plot suggests that there is seasonal variation with a very slight downward trend 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
head(GaringerOzone)
```

```
##         Date Daily.Max.8.hour.Ozone.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
```

```
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
```

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

summary(Garinger_Ozone_clean$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
```

> Answer: We didn't use the piecewise constant b/c there is high variation between dates so taking the nearest neighbor as taken in piecewise constant would flatten that variation. Spline wouldn't be appropriate because the trend line is linear so using a quadratic function wouldn't fit it as well. The linear interpolation method is the most appropriate by assuming that the missing data fall between the last and the next measurent with a straight line drawn between them.

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 <- Garinger_Ozone_clean %>%
  mutate(Month=month(Date))%>%
  mutate(Year=year(Date))%>%
  group_by(Year, Month) %>%
  summarise(Mean_ozone_concentration =
            mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date=ymd(paste(Year,"-",Month,"-01")))
class(GaringerOzone.monthly$Date)
```

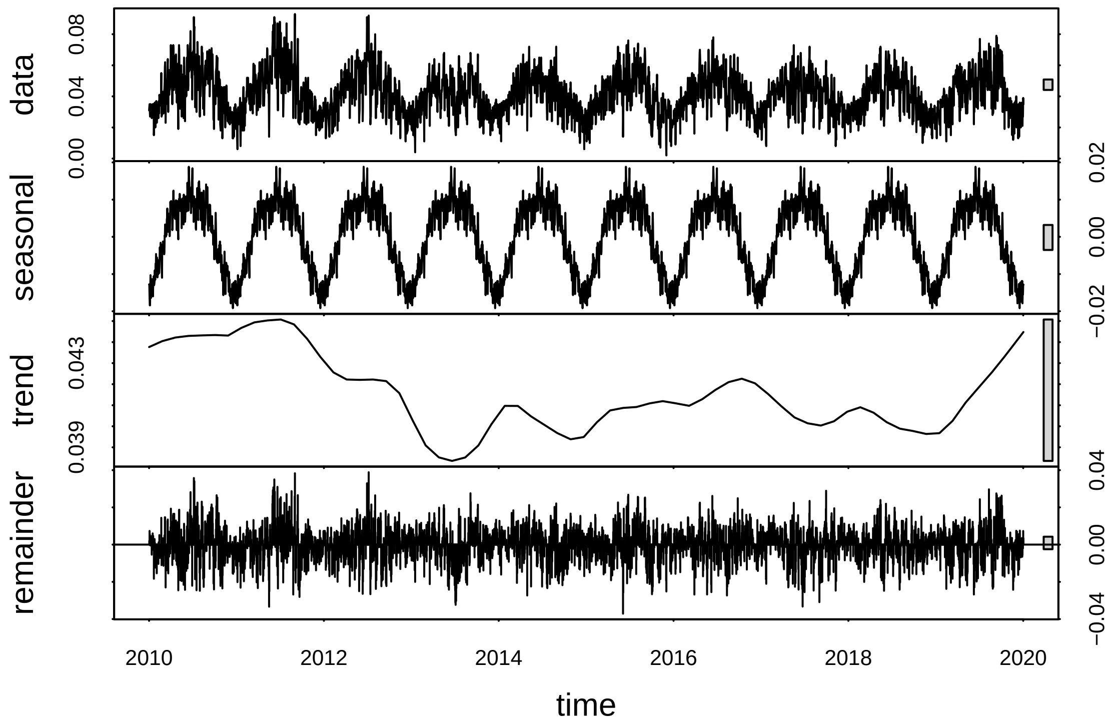
```
## [1] "Date"
```

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.

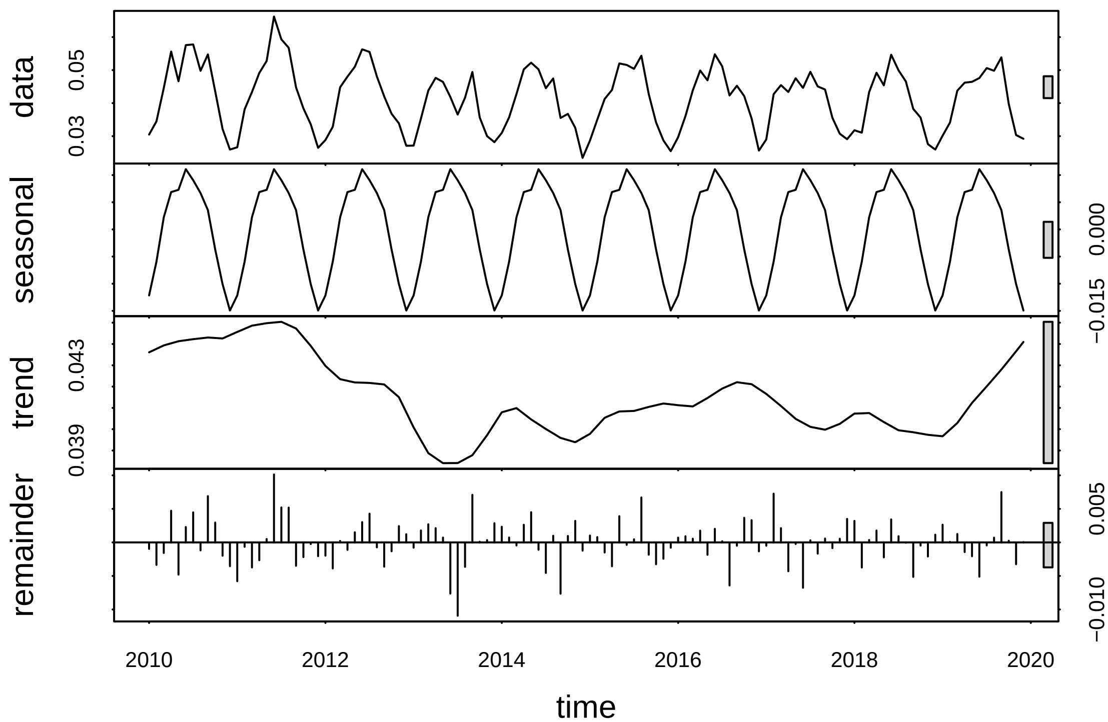
```
#10
GaringerOzone.daily.ts <- ts(Garinger_Ozone_clean$Daily.Max.8.hour.Ozone.Concentration, start = c(2010,

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_ozone_concentration,start=c(2010,1),frequency
```

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

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

Garinger.mean.decomposed <- stl(GaringerOzone.monthly.ts,s.window="periodic")

plot(Garinger.daily.decomposed)
```




```
plot(Garinger.mean.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
Ozone_trend_smk<- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

Ozone_trend_smk
```

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

```
summary(Ozone_trend_smk)
```

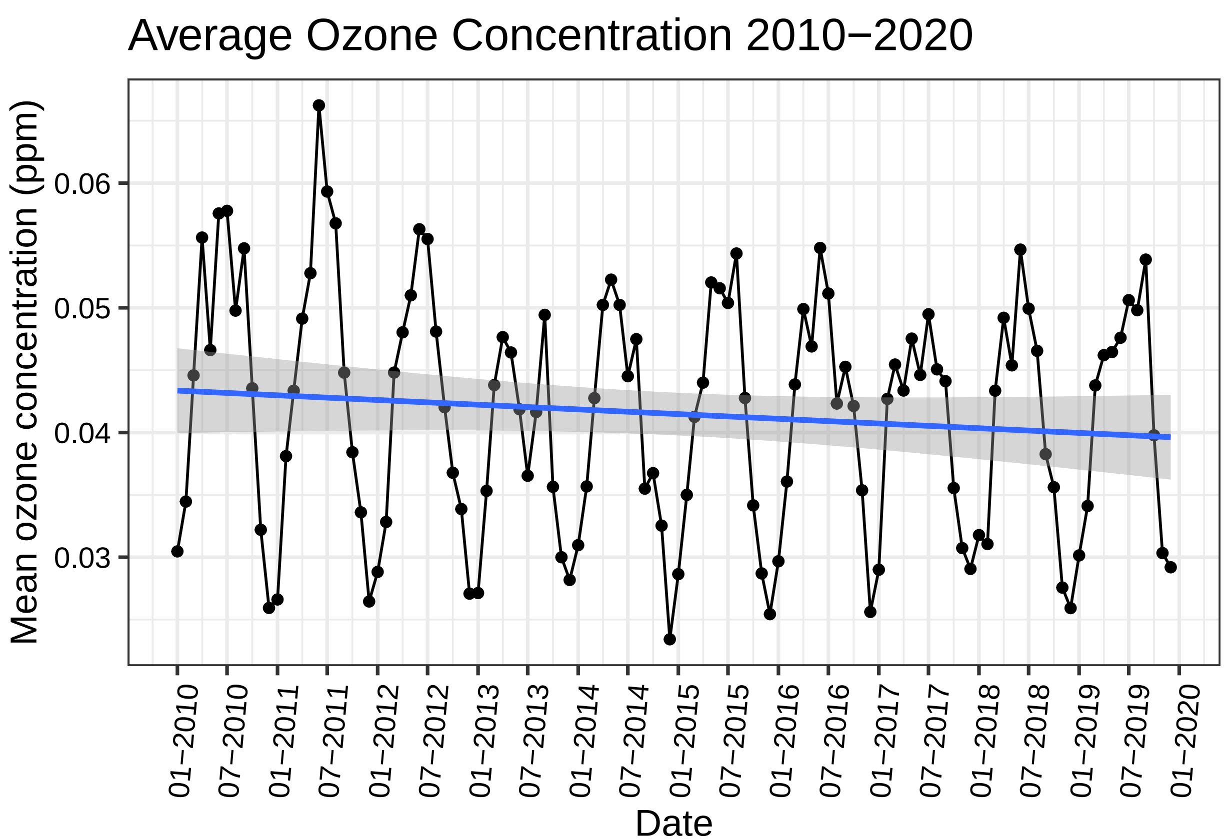
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: A seasonal Mann-Kendall test is the only one of the four we discussed that takes seasonality into account. According to our decomposed time series plotted, a seasonal pattern is suggested.

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

 Monthly_ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_ozone_concentration)) +
  geom_point() +
  geom_line() +
  labs(title="Average Ozone Concentration 2010-2020")+
  ylab("Mean ozone concentration (ppm)") +
  geom_smooth( method = lm )+
    scale_x_date(labels = date_format("%m-%Y"),
date_breaks = "6 months") + theme(axis.text.x = element_text(angle = 85,
hjust = 1))

print(Monthly_ozone_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: The research question asked us "Have ozone concentrations changed over the 2010s at this station?" From the graph we can see that there is slightly less seasonal variation as we move later into the 2010s, with lower highs. There is also a very slight negative trend over the decade

(The seasonal Mann Kendall test tells us that there is a minor negative trend (-0.143) with slight significance looking at a p-score not much smaller than 0.05 (p-value=0.0467))

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
Garinger_Ozone_nonseasonal <- as.data.frame(Garinger.mean.decomposed$time.series[,2:3])

Garinger_Ozone_nonseasonal <- mutate(Garinger_Ozone_nonseasonal,
        Observed = Garinger_Ozone_nonseasonal$trend,
        Date = GaringerOzone.monthly$Date)
Garinger_Ozone_nonseasonal <- Garinger_Ozone_nonseasonal %>%
  select(Observed,Date)
Garinger_nonseasonal.ts <- ts(Garinger_Ozone_nonseasonal$Observed, start=c(2010,1),frequency=12)

#16

Ozone_trend_nonseasonal_smk<- Kendall::MannKendall(Garinger_nonseasonal.ts)

Ozone_trend_nonseasonal_smk
```

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

```
summary(Ozone_trend_nonseasonal_smk)
```

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

Answer:The Mann:Kendall Test run on the monthly ozone data with seasonality extracted showed a slightly stronger negative trend, with a tau score of -0.269 compared to the seasonal data that had a tau score of -0.143. The non-seasonal data also has a much stronger significance with a p-value far below .05 compared to our seasonal p-value that was barely below .05 (.0467).