Assignment 7: Time Series Analysis

Stephanie Kinser

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] "C:/Users/skins/Documents/EDA/Environmental_Data_Analytics_2022/Assignments"
```

```
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(trend)
library(dplyr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

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

```
A7_theme <- theme_light(base_size = 12)+
   theme(axis.text = element_text(color = "black"),
         legend.position = "right", panel.grid.minor = element_blank())

theme_set(A7_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
#Garinger_files <- setwd("./Data/Raw/Ozone_TimeSeries/")
getwd()
```

```
## [1] "C:/Users/skins/Documents/EDA/Environmental_Data_Analytics_2022/Assignments"
```

```
# Get the files names
o2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = TR
o2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = TR
o2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = TR
o2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = TR
o2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = TR
o2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = TR
o2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = TR
o2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = TR
o2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = TR
o2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = TR

GaringerOzone <- rbind(o2010, o2011, o2012, o2013, o2014, o2015, o2016, o2017, o2018, o2019)
```

```
# files = list.files(pattern="*.csv")
#
# # First apply read.csv, then rbind
# GaringerOzone = do.call(rbind, lapply(files, function(x) read.csv(x, stringsAsFactors = TRUE)))
```

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_tidy <- GaringerOzone %>%
  select(Date,
         Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE) %>%
  rename(Ozone = Daily.Max.8.hour.Ozone.Concentration) %>%
  rename(AQI = DAILY_AQI_VALUE)


# 5
Days = as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day")) %>% #by
  setNames(c("Date")) #renaming column to date
class(Days$Date)
```

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

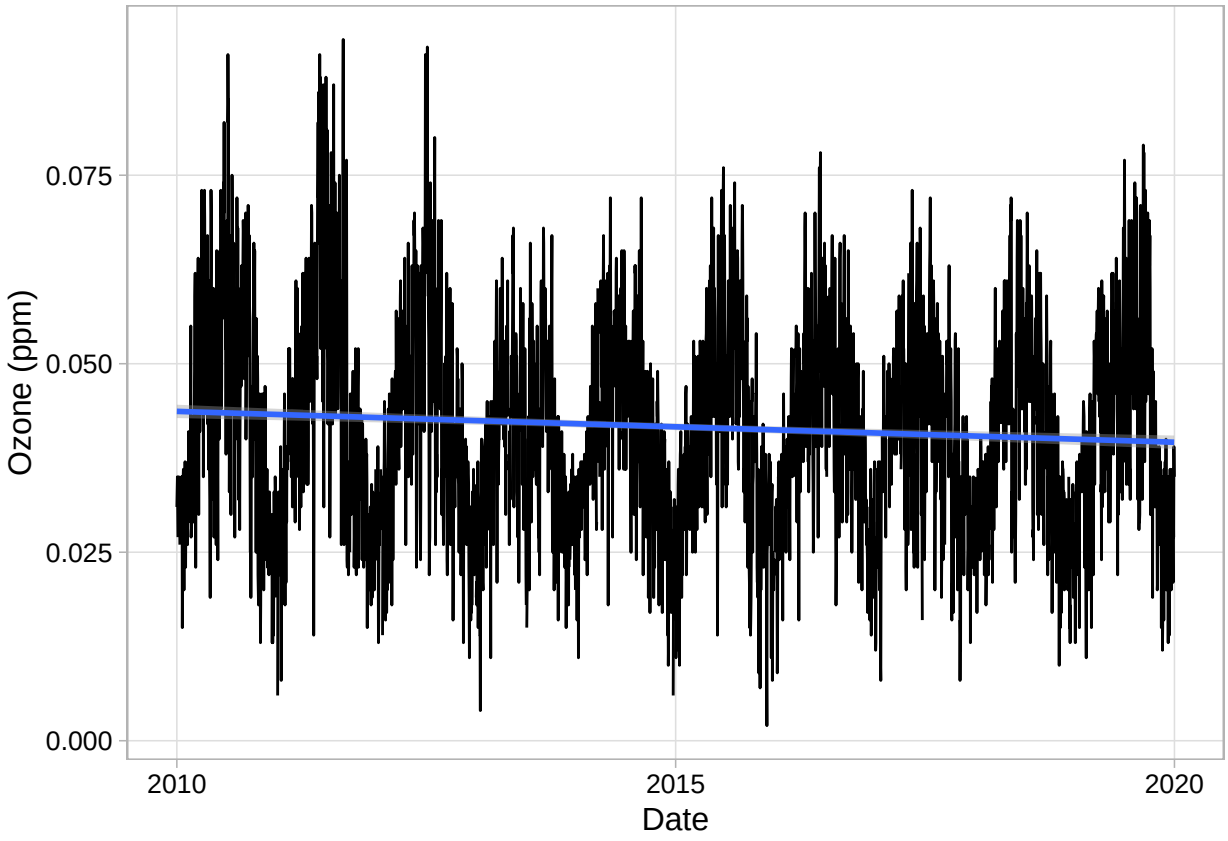
```
# 6
GaringerOzone <- left_join(Days, GaringerOzone_tidy, by = "Date") #Days must come first to join with it
```

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
ozone_plot <- ggplot(GaringerOzone, aes(x = Date, y = Ozone)) +
  geom_line() +
  geom_smooth(method = lm)+
  labs(x = "Date", y = "Ozone (ppm)")
ozone_plot
```

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



Answer: The plot suggests a seasonal trend for ozone and also that ozone concentrations have decreased slightly from 2010 to 2019.

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)
```

```
##       Date                Ozone              AQI
##  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 <-
  GaringerOzone %>%
  mutate(Ozone = zoo::na.approx(Ozone))
```

Answer: We did not use piecewise constant because much of the missing data's "nearest neighbor" is also missing, so the observations cannot be easily filled in with data from observations on nearby dates. We do not use a spline function because our data does not show a quadratic trend.

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
#new monthly data frame
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date)) %>%
  group_by(year, month) %>%
  summarise(mean_Ozone = mean(Ozone))
```

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

```
#new date column
GaringerOzone.monthly$monthyear <- paste("01", GaringerOzone.monthly$month, GaringerOzone.monthly$year,

#change to Date format
GaringerOzone.monthly$monthyear <- as.Date(GaringerOzone.monthly$monthyear, format = "%d-%m-%Y" )
class(GaringerOzone.monthly$monthyear)
```

```
## [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
fday <- day(first(GaringerOzone$Date)) #probably assumes first day of the month
fmonth <- month(first(GaringerOzone$Date))
fyear <- year(first(GaringerOzone$Date))
```

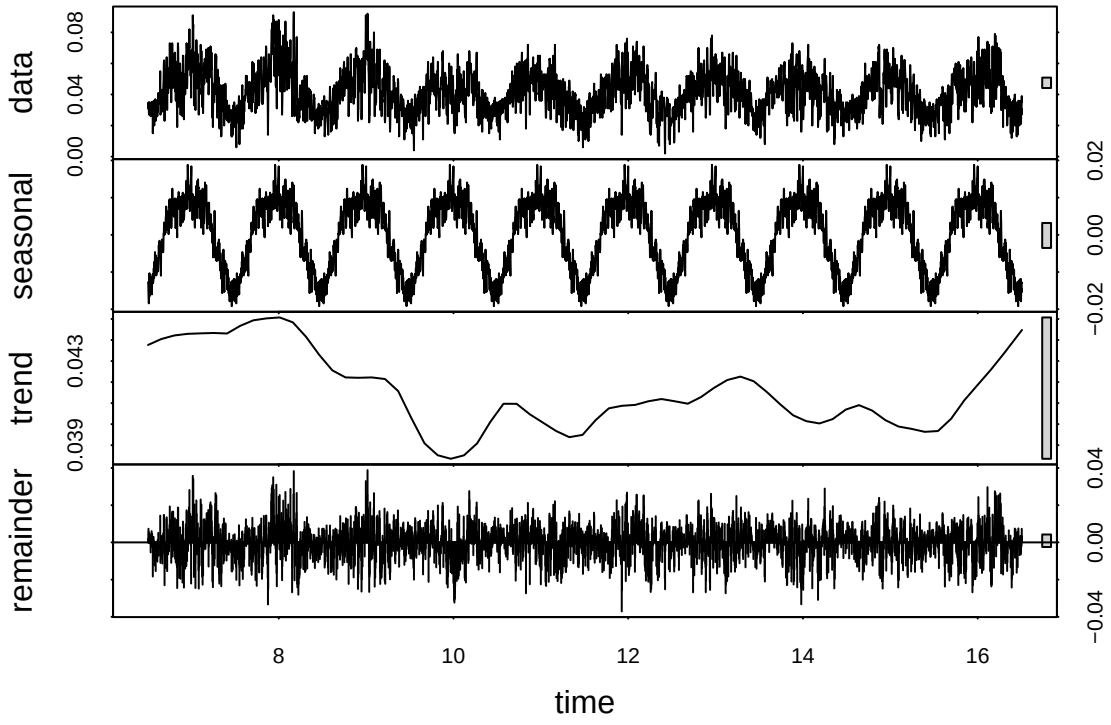
```
GaringerOzone.daily.ts <- ts(GaringerOzone$Ozone, start = c(fmonth, fyear), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_Ozone, start = c(fmonth, fyear), frequency =
```

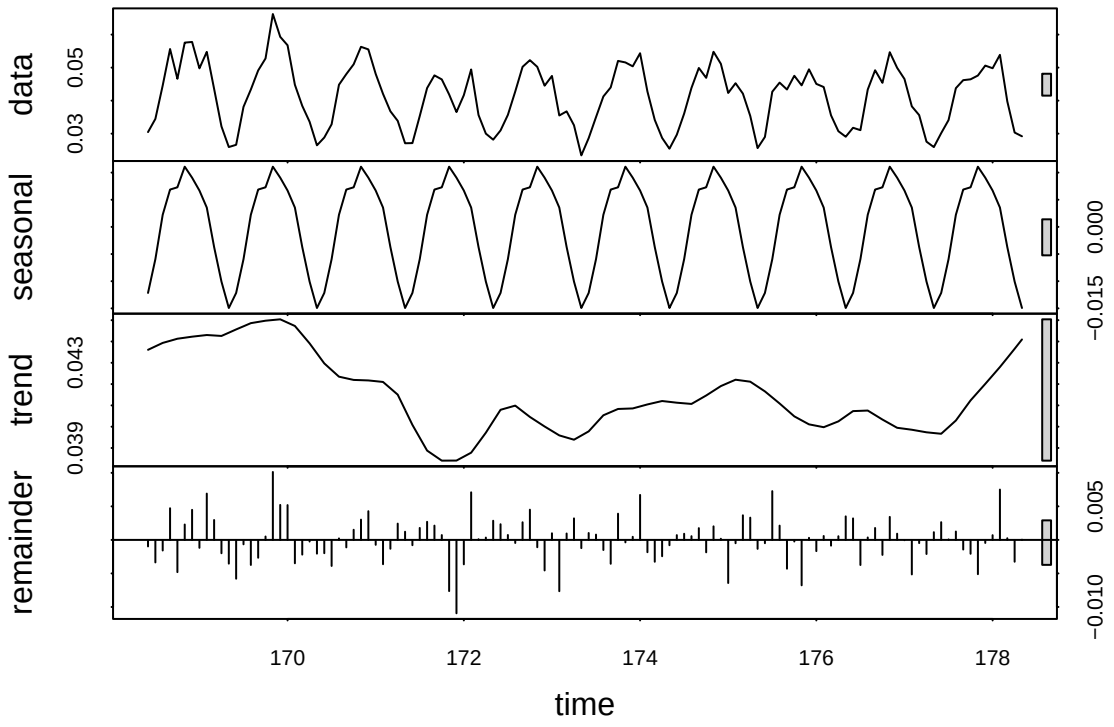
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

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




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

#Seasonal Mann Kendall
ozone_daily_trend <- Kendall::SeasonalMannKendall(GaringerOzone.daily.ts)

# Inspect results
ozone_daily_trend
```

```
## tau = -0.0456, 2-sided pvalue =0.00051075
```

```
summary(ozone_daily_trend)
```

```
## Score =  -739 , Var(Score) = 45223.67
## denominator =  16213.86
## tau = -0.0456, 2-sided pvalue =0.00051075
```

```
#Seasonal Mann Kendall
ozone_monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
ozone_monthly_trend
```

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

```
summary(ozone_monthly_trend)
```

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

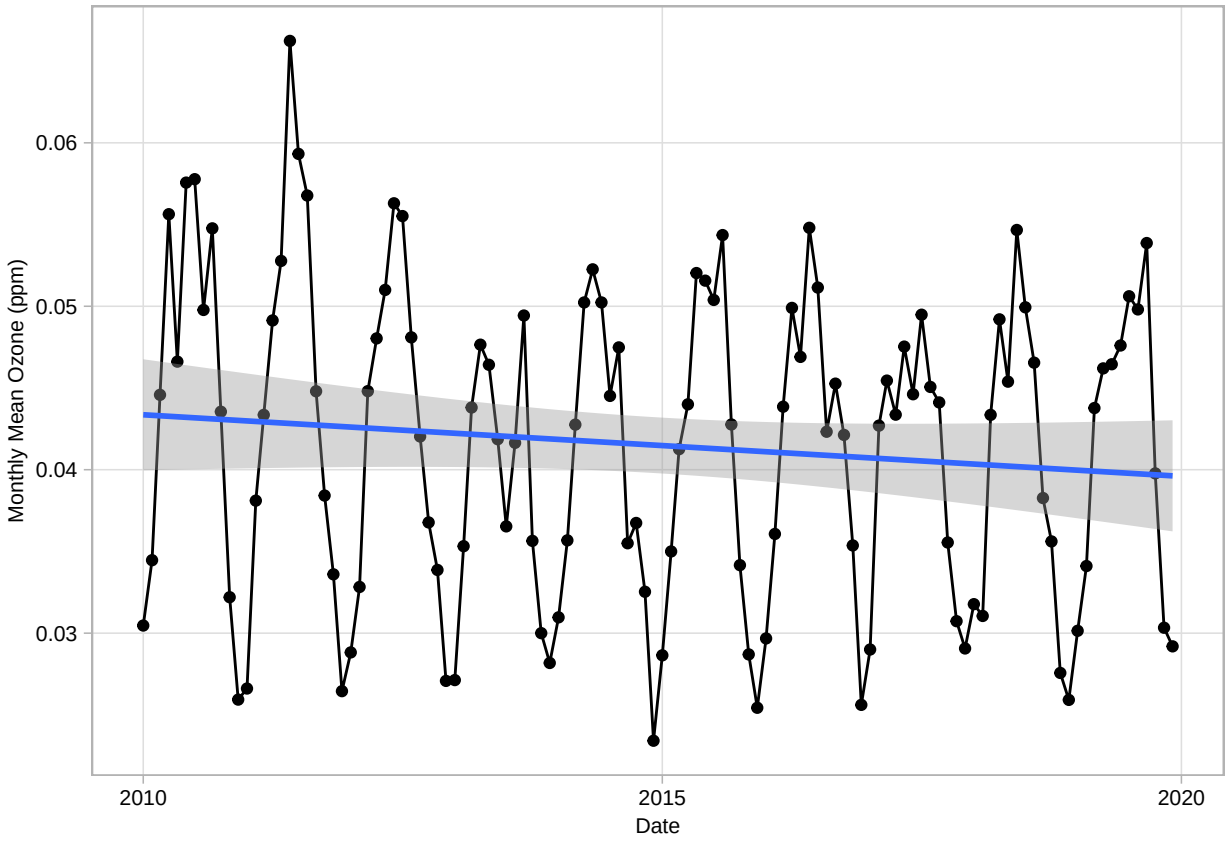
Answer: The seasonal Mann-Kendall is the most appropriate test because the ozone concentrations shows seasonal variability and the seasonal Mann-Kendall is the only test that can account for this seasonal variability.

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 = monthyear, y = mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Monthly Mean Ozone (ppm)") +
  geom_smooth( method = lm )+
  theme(axis.text = element_text(size = 8), axis.title = element_text(size = 8))
print(ozone_monthly_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 graph and seasonal Mann-Kendall test show a gentle downward sloping trend for ozone concentrations from 2010-2019. The monthly mean concentration of ozone shows a stronger downward sloping trend (tau = -0.143, p-value = 0.047) than the daily observations (tau = -0.046, p-value = 0.0005). However, both seasonal Mann-Kendall analyses and the plot support decreasing concentrations and are statistically significant.

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
#extracting the nonseasonal components of ts
ozone_nonseasonal <- as.data.frame(ozone_monthly_decomp$time.series[,1:3])

#combining the nonseasonal components
ozone_nonseasonal <- ozone_nonseasonal %>%
  mutate(nonseasonal = trend + remainder) %>%
  select(nonseasonal)
```

```r
#adding data column to new df
ozone_nonseasonal$Date <- GaringerOzone.monthly$monthyear


#16
#new ts on nonseasonal data
ozone_nonseasonal_ts <- ts(ozone_nonseasonal$nonseasonal, start = c(2010-01-01), end = c(2019-01-01), fr

#Mann Kendall test
ozone_nonseasonal <- Kendall::MannKendall(ozone_nonseasonal_ts)

# Inspect results
ozone_nonseasonal
```

```
## tau = -0.206, 2-sided pvalue =0.0015052
```

```r
summary(ozone_nonseasonal)
```

```
## Score =  -1213 , Var(Score) = 145841
## denominator =  5885.5
## tau = -0.206, 2-sided pvalue =0.0015052
```

Answer: The Mann-Kendall test on the nonseasonal monthly ozone concentrations shows a stronger downward trend (tau = -0.206, p-value = 0.0015) than the seasonal Mann-Kendall test on the complete data (tau = -0.143, p-value = 0.047). The smaller tau value indicates a more significant negative trend in the nonseasonal data.