Assignment 8: Time Series Analysis

Fiona Price

Fall 2024

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

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. Import data in bulk and combine into a single dataframe.
folder <- here('Data','Raw','Ozone_TimeSeries')
files <- dir(folder,pattern = 'EPAair_O3*.*',full.names = T) #Looks for the
#name pattern of EPAair_O3

GaringerOzone <- files %>%
  map(read.csv) %>%
  reduce(rbind) #Combines multiple data frames by binding rows together
```

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

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

```
# 4. Wrangle your dataset so that it only contains the columns Date,
#Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.
GaringerOzone_wrangled <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5. Create a new data frame that contains a sequence of dates from 2010-01-01
#to 2019-12-31.
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
                          by = "day"))
#Rename the column to Date.
colnames(Days) <- c("Date")

# 6. Use a `left_join` to combine the data frames.
GaringerOzoneCombined <- left_join(Days, GaringerOzone_wrangled)
```

```
## Joining with 'by = join_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. Create a line plot depicting ozone concentrations over time.
ozone_plot <- ggplot(GaringerOzoneCombined,
       aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() + #Specify line graph
  geom_smooth(method = "lm") + #Add smoothed trend line
  labs(title = "Ozone Concentrations Over Time",
```

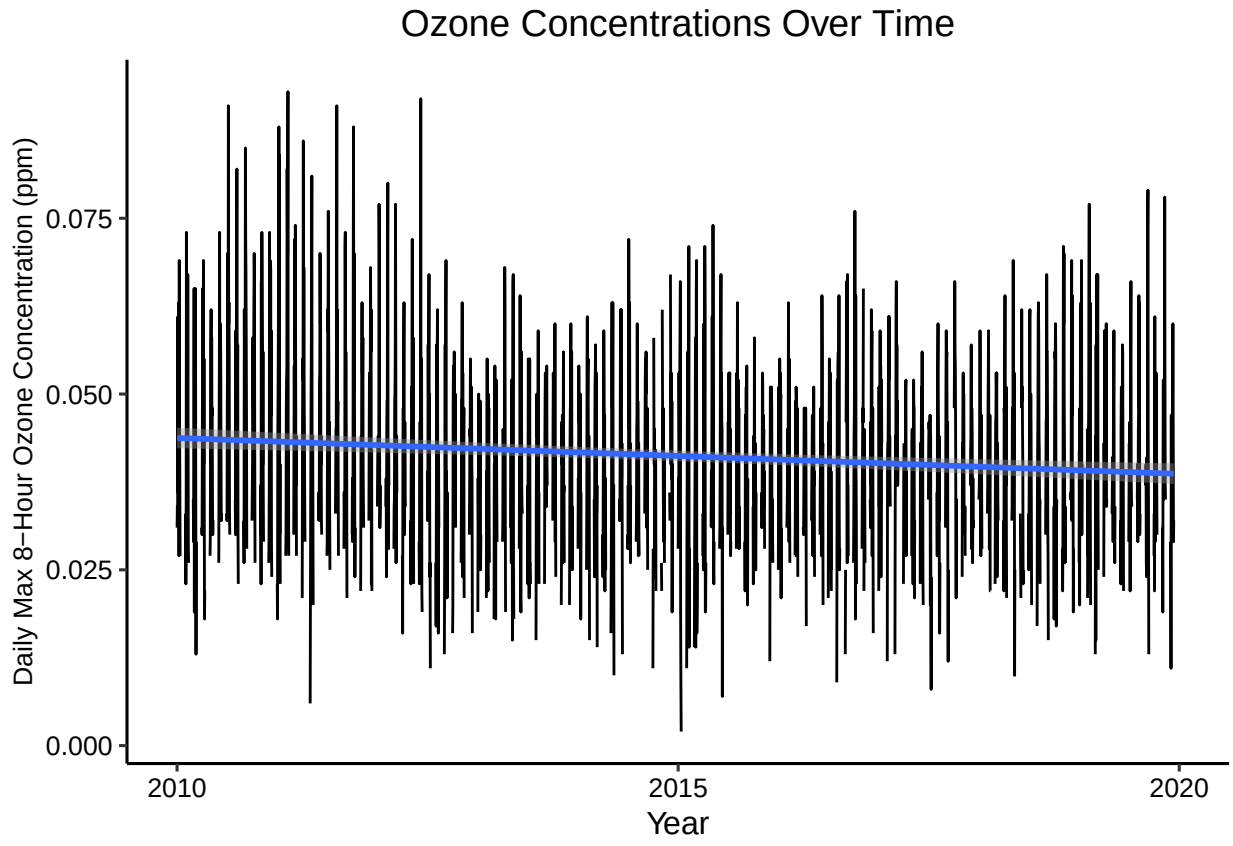
```
       x = "Year",
       y = "Daily Max 8-Hour Ozone Concentration (ppm)") + #Add labels
  mytheme + #Add my theme
  theme(axis.title.y = element_text(size = 10)) #Change size of y-axis label

plot(ozone_plot)
```

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

```
## Warning: Removed 2231 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 19 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Answer: Yes, my plot suggests a downward trend in ozone concentration 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?

```r
#8. Use a linear interpolation to fill in missing daily data for ozone
#concentration.
summary(GaringerOzoneCombined$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0020  0.0310  0.0400  0.0412  0.0500  0.0930    2231
```

```r
#There are 2231 NAs.

# Adding new column with no missing observations.
Garinger_data_clean <-
  GaringerOzoneCombined %>%
  mutate(Ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration,
                                      na.rm = FALSE, rule = 2))
#I had to add na.rm = FALSE to prevent the NAs from being omitted initially.
#The rule = 2 allows the function to extrapolate for trailing NAs.
#Check for NAs.
summary(Garinger_data_clean$Ozone.clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.02537 0.03005 0.03260 0.03600 0.09300
```

```r
#No more NAs, and still have 3652 observations!
```

Answer: A piecewise constant interpolation method assumes that missing data are equal to the measurement made nearest to that date; essentially, it assumes that the value remainds constant over time. However, our value is changing over time, so a piecewise constant fails to capture that. A spline method would also not be the best choice for our data set because our data shows a linear trend over time, while the splien uses a quadratic function to interpolate.

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)

```r
#9. Create a new data frame called `GaringerOzone.monthly` that contains
#aggregated data: mean ozone concentrations for each month.
GaringerOzone.monthly <- Garinger_data_clean %>%
  mutate(Year = year(Date)) %>%
  mutate(Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE),
            .groups = 'drop')

#Create a new Date column with each month-year combination being set as
#the first day of the month.
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(Year, month(Month), 1))
```

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.

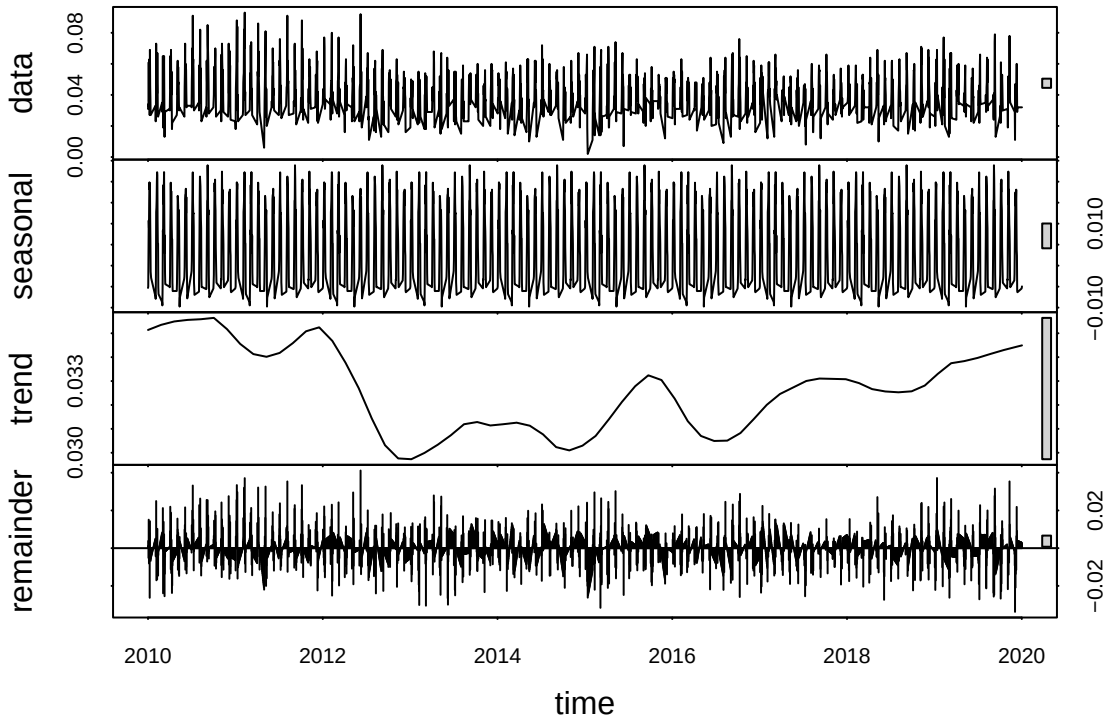
```r
#10. Generate a time series object based on daily observations.
GaringerOzone.daily.ts <- ts(Garinger_data_clean$Ozone.clean,
                             start = c(2010), frequency = 365)

# Generate a time series object based on monthly observations.
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
                             start = c(2010), frequency = 12)
```

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

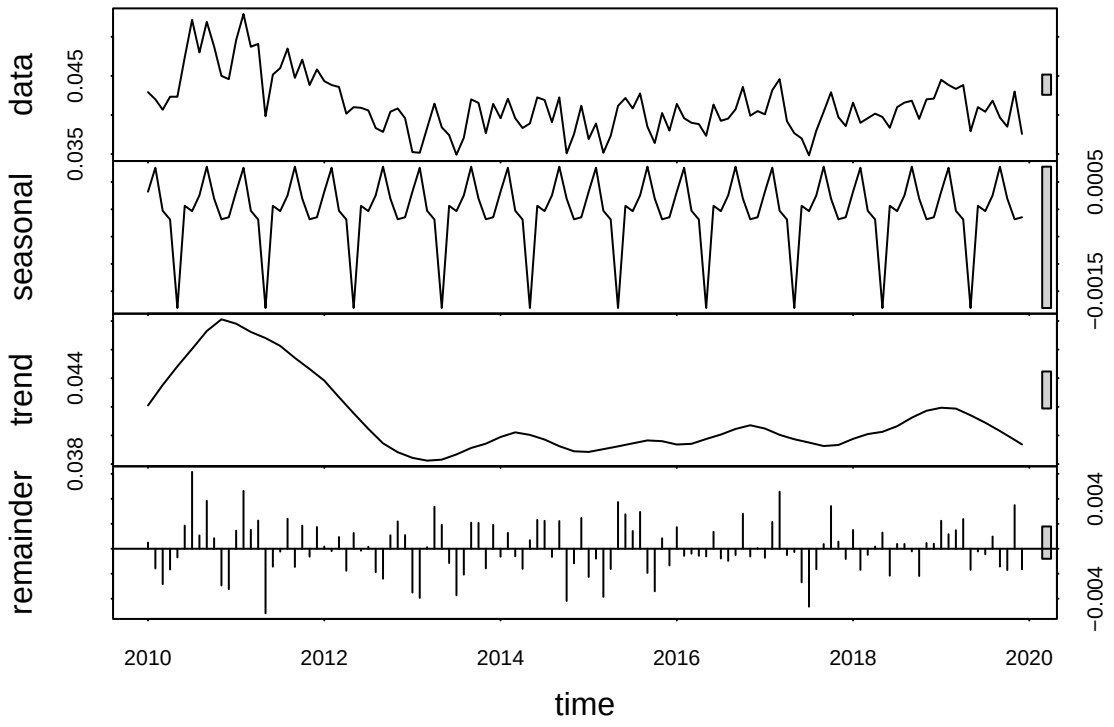
```r
#11. Generate the decomposition for daily data.
GaringerOzone.daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

# Visualize the decomposed series.
plot(GaringerOzone.daily_Decomposed)
```




```r
# Generate the decomposition for monthly data.
GaringerOzone.monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# Visualize the decomposed series.
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. Run a monotonic trend analysis for the monthly Ozone series.
monthly_data_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly_data_trend
```

```
## tau = -0.251, 2-sided pvalue =0.00048451
```

```
summary(monthly_data_trend)
```

```
## Score =  -135 , Var(Score) = 1497
## denominator =  538.4916
## tau = -0.251, 2-sided pvalue =0.00048451
```

```
# We can use another Seasonal Mann-Kendall function to extract data for specific
# seasons as well.
monthly_data_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
summary(monthly_data_trend2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
```

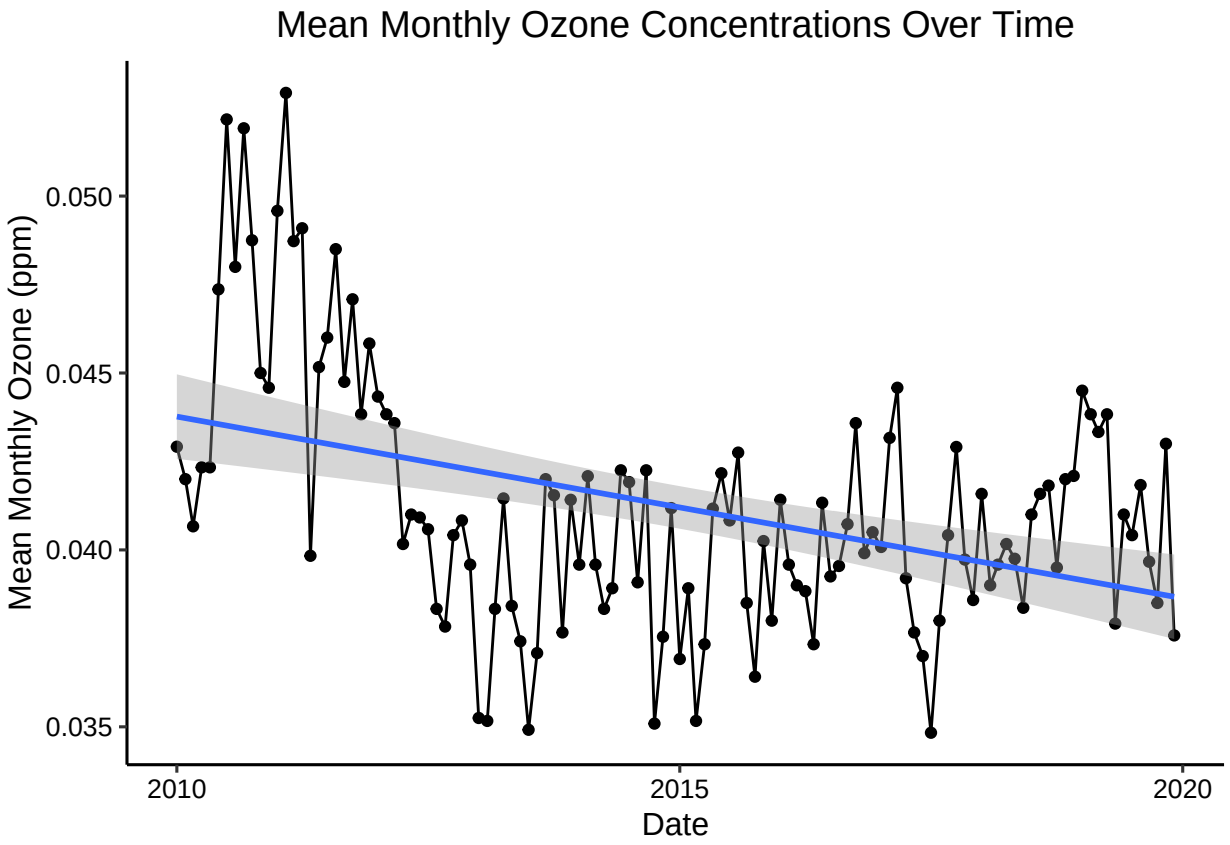
```
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0    1  125  0.022  0.000 1.000000
## Season 2:   S = 0   -2  124 -0.045 -0.090 0.928444
## Season 3:   S = 0   -4  124 -0.090 -0.269 0.787616
## Season 4:   S = 0   -7  124 -0.157 -0.539 0.590014
## Season 5:   S = 0  -19  125 -0.422 -1.610 0.107405
## Season 6:   S = 0  -21  125 -0.467 -1.789 0.073638 .
## Season 7:   S = 0  -19  125 -0.422 -1.610 0.107405
## Season 8:   S = 0   -3  125 -0.067 -0.179 0.858028
## Season 9:   S = 0  -17  125 -0.378 -1.431 0.152406
## Season 10:   S = 0 -17  125 -0.378 -1.431 0.152406
## Season 11:   S = 0  -7  125 -0.156 -0.537 0.591505
## Season 12:   S = 0 -21  125 -0.467 -1.789 0.073638 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: Because our trend is not exactly linear, we need to use a different trend analysis method. The Mann-Kendall test does not assume anything about the linearity or distribution of the data, making it a better fit in our case. Furthermore, it allows for missing data (which we have). Specifically, the seasonal Mann-Kendall test should be use to consider periodic fluctuations due to seasonality.

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. Create a plot depicting mean monthly ozone concentrations over time.
monthly_ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone)) +
  geom_point() + #specify scatter plot
  geom_line() + #add lines to connect points
  labs(title = "Mean Monthly Ozone Concentrations Over Time",
       y = "Mean Monthly Ozone (ppm)") + #edit titles
  geom_smooth(method = lm) + #add trend line
  mytheme #add my theme
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: Mean monthly ozone concentrations have decreased over the 2010s at this station. This trend is statistically significant, meaning the null hypothesis (that there is no trend over time) is rejected (pvalue =0.00048451). The trend is moderately negative based on Kendall's tau statistic, which ranges from -1 to 1 (-0.251 for our data). There is also a fair bit of variation in the plot, which is also reflected in our variance score (1497).

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. Extract the components and turn them into a data frame.
GaringerMonthly_Components <- as.data.frame(GaringerOzone.monthly_Decomposed$time.series)

# Change column names
GaringerMonthly_Components <- mutate(GaringerMonthly_Components,
        Observed = GaringerOzone.monthly$Mean_Ozone,
        Date = GaringerOzone.monthly$Date)

# Select specific time series components (note we are NOT selecting seasonality).
```

```r
GaringerMonthly_Components <- GaringerMonthly_Components %>%
  select(trend, remainder, Observed, Date)


#16. Run the Mann Kendall test on the non-seasonal Ozone monthly series.
non.seasonal.ts <- ts(GaringerMonthly_Components$Observed,
                              start = c(2010), frequency = 12)

non.seasonal <- Kendall::MannKendall(non.seasonal.ts)
# Show results
summary(non.seasonal)
```

```
## Score =  -1535 , Var(Score) = 194311.7
## denominator =  7118.467
## tau = -0.216, 2-sided pvalue =0.00050146
```

Answer: The non-seasonal trend in monthly ozone concentrations is less negative (though it is still negative) than the seasonal one, as seen in the higher tau statistic (-0.216 vs. -0.251). Additionally, there is more variation (194311.7 vs. 1497). Both of these trends are statistically significant. Overall, for both the seasonal and non-seasonal monthly ozone time series, ozone concentrations decrease over time.