Assignment 8: Time Series Analysis

Asreeta Ushasri

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

```r
#Check working directory

#getwd()

#Load packages

#install.packages("tidyverse")
#install.packages("zoo")
#install.packages("lubridate")
#install.packages("trend")
#install.packages("Kendall")
#install.packages("tseries")
#install.packages("cowplot")
#install.packages("viridisLite")

library(tidyverse)
library(here)
library(zoo)
library(lubridate)
library(trend)
library(Kendall)
library(tseries)
library(cowplot)
library(viridis)
library(RColorBrewer)
library(colormap)
library(ggthemes)

#Set custom theme

my_theme <- theme_classic() + theme(

#Text Color
  plot.title = element_text(color = "black"),
  axis.title.x = element_text(color = "black"),
  axis.title.y = element_text(color = "black"),
  axis.text = element_text(color = "black"),

#Line Color
  panel.grid.major = element_line(color = "lightgray"),
  panel.grid.minor = element_line(color = "lightgray"),

#Rectangle Element
  panel.background = element_rect(color = "white"),
  legend.key = element_rect(color = "white"),

#Legend Position
  legend.position = 'bottom',
complete = TRUE)

theme_set(my_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.

```r
#1 Import data sets

Ozone_2019_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2018_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2017_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)
```

```r
Ozone_2016_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2015_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2014_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2013_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2012_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2011_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)

Ozone_2010_Raw <- read.csv(
  here("Data", "Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)

#Combine data from different years into one data frame

GaringerOzone_Raw <- rbind(Ozone_2010_Raw, Ozone_2011_Raw, Ozone_2012_Raw,
                       Ozone_2013_Raw, Ozone_2014_Raw, Ozone_2015_Raw,
                       Ozone_2016_Raw, Ozone_2017_Raw, Ozone_2018_Raw,
                       Ozone_2019_Raw)

#view(GaringerOzone_Raw)
```

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.

```r
#3 Set date column as date class

class(GaringerOzone_Raw$Date)
```

```
## [1] "factor"
```

```r
GaringerOzone_Raw$Date <- as.Date(GaringerOzone_Raw$Date, format = "%m/%d/%Y")

class(GaringerOzone_Raw$Date)
```

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

```r
#view(GaringerOzone_Raw)

#4 Select needed columns in data

GaringerOzone_Raw_Columns <- select(GaringerOzone_Raw, "Date",
                                    "Daily.Max.8.hour.Ozone.Concentration",
                                    "DAILY_AQI_VALUE")

#view(GaringerOzone_Raw_Columns)

#5 Create a new data frame with a date sequence

Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"), by = "day"))

colnames(Days) <- "Date"

#view(Days)

#6 Combine Data Frames

GaringerOzone <- left_join(x = Days, y = GaringerOzone_Raw_Columns,
                           by = "Date")

#view(GaringerOzone)
```

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?

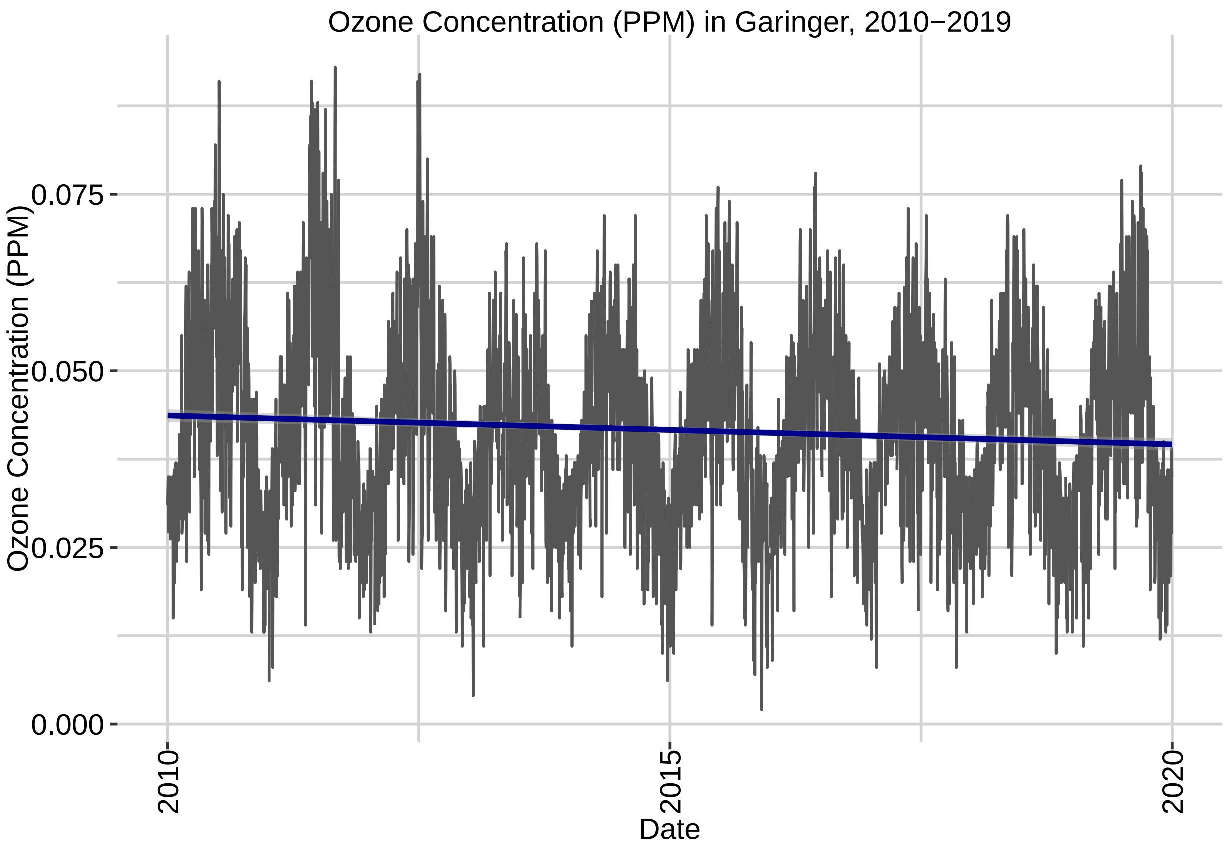
```
#7 Create a line plot for ppm analysis

GaringerOzone_PPM <- ggplot(data = GaringerOzone,
                            aes(Date, Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(colour = "#555") + labs(
    title = "Ozone Concentration (PPM) in Garinger, 2010-2019",
    x = "Date",
    y = "Ozone Concentration (PPM)") +
  theme(axis.title.y =
             element_text(angle = 90)) +
  theme(axis.text.x =
             element_text(angle = 90)) +
  geom_smooth(method = "lm", colour = "darkblue")

print(GaringerOzone_PPM)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```



Answer: The plot suggests there is seasonal variation in the ozone concentration in Garinger, because within each year the graph increases and decreases. Over an entire decade, this trend produces a cyclical nature in the graph. Scientifically, ground-level ozone occurs from a reaction between NOx (nitrogen oxides), sunlight, and VOCs (volatile organic compounds). Because sunlight is a component of this reaction, ozone concentrations tend to increase in the summer when daylight lasts longer. This graph could correlate to this seasonal variation. However, this hypothesis must be proven using time series analysis tests.

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

#view(GaringerOzone_Clean)
```

Answer: The piecewise constant interpolation assumes the missing data is the same as the nearest neighbor. This approach would not work in this case, because ozone concentrations have daily variations, and we cannot assume the ozone concentration would be the same two days in a row. The spline interpolation works best for non-linear trends with a quadratic technique. In general, the difference between two days can be mapped with a line instead of a non-linear function. A linear interpolation assigns a value in between the two nearest neighbors for any missing data based on a linear trend between the two neighbors. Therefore, a linear interpolation fits this data best.

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 Create a data frame for monthly concentrations

GaringerOzone.monthly <-
  GaringerOzone_Clean %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
         group_by(Year, Month) %>%
    summarise(Mean.Ozone = mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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

```
#view(GaringerOzone.monthly)
```

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year = Year, month = Month, day = "01"))

#view(GaringerOzone.monthly)
```

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 Create two time series

daily_month <- month(first(GaringerOzone_Clean$Date))
daily_year <- year(first(GaringerOzone_Clean$Date))

GaringerOzone.daily.ts <-
  ts(GaringerOzone_Clean$Daily.Max.8.hour.Ozone.Concentration.clean,
     start=c(daily_year,daily_month),
                   frequency=365)

#view(GaringerOzone.daily.ts)


monthly_month <- month(first(GaringerOzone.monthly$Date))
monthly_year <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Ozone,
                               start=c(monthly_year,monthly_month),
                   frequency=12)

#view(GaringerOzone.monthly.ts)
```

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

```
#11 Decompose and plot results

Garinger_Daily_Decomp <-
  stl(GaringerOzone.daily.ts,s.window = "periodic")
dev.new(width = 10, height = 7)
plot(Garinger_Daily_Decomp,
     main = "Daily Garinger Ozone Concentration (PPM)")


Garinger_Monthly_Decomp <-
  stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(Garinger_Monthly_Decomp,
     main = "Monthly Garinger Ozone Concentration (PPM)")
```

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 Monotonic Trend Analysis for Monthly Ozone

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

> Answer: The Seasonal Mann-Kendall test was created to handle seasonal variation. The ozone levels showcase clear cyclical trend within each year in our first graph, indicating seasonal trends. The other monotonic analyses (linear regression, Mann-Kendall, Spearman Rho etc.) cannot accommodate seasonality in their trend test. Thus, the Seasonal Mann-Kendall test is the only appropriate method for this data.

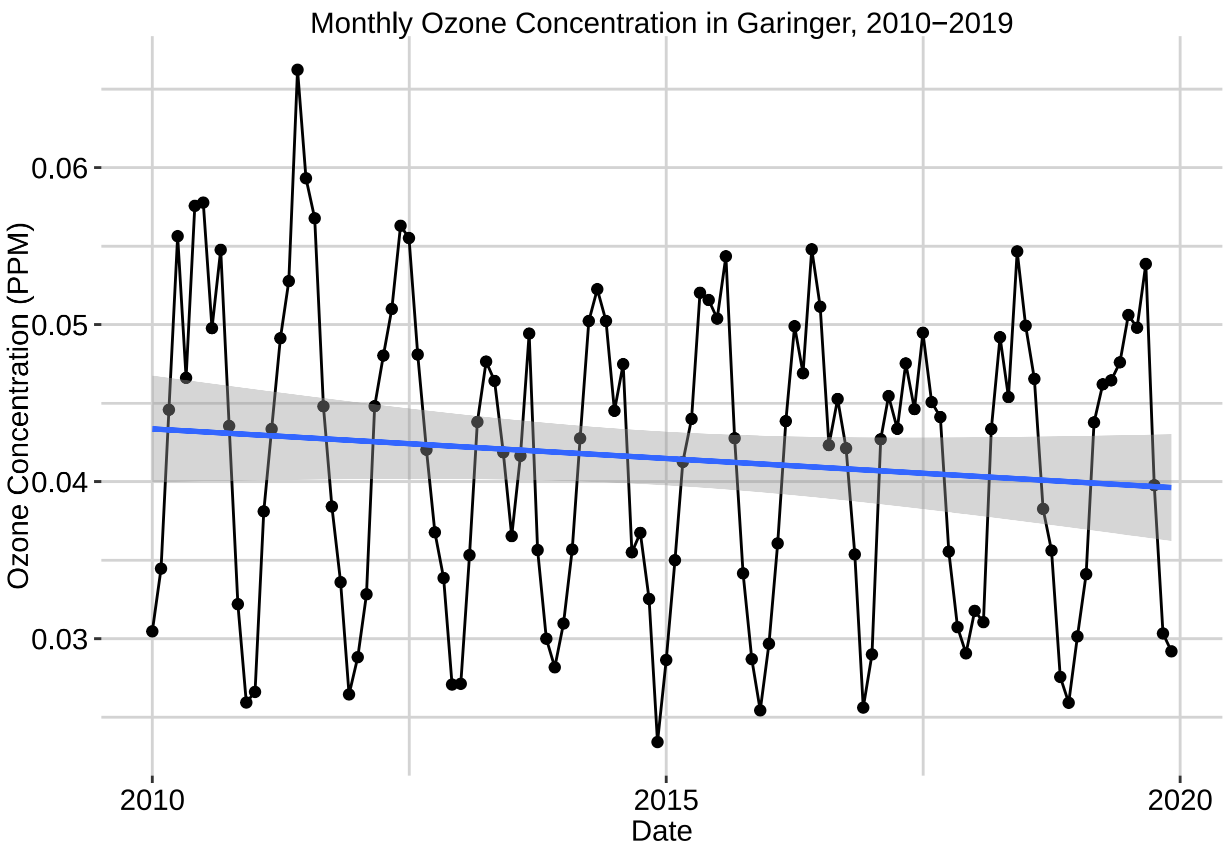
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13 Plot Monthly Trend Analysis

Garinger_Ozone_Trend_Monthly_Plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Ozone)) +
  geom_point() +
  geom_line() +  labs(
    title = "Monthly Ozone Concentration in Garinger, 2010-2019",
    x = "Date",
    y = "Ozone Concentration (PPM)") +
  theme(axis.title.y =
             element_text(angle = 90)) +
  geom_smooth( method = lm )

print(Garinger_Ozone_Trend_Monthly_Plot)
```

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



```
#Print statistical results
Garinger_Ozone_Trend_Monthly
```

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

```
summary(Garinger_Ozone_Trend_Monthly)
```

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

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 asks if ozone concentration has changed at this station from 2010 to 2019. According to the linear trend depicted on this graph, the ozone concentration has slightly decreased over this decade. The ozone concentration trendline dropped from roughly 0.44 to 0.39 PPM from 2010 to 2019. This depicts a gradual decrease in ozone concentration at this station.

    The two-sided p-value from the monotonic trend analysis was 0.046, which is statistically significant. Therefore, we would reject the null hypothesis. The null hypothesis is typically that no relationship exists between the variables in question. If we reject the null hypothesis, that means there is a trend between date and ozone concentration on a monthly basis.

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.

```r
#15 Removing seasonality

#Ensure type is data frame

Garinger_Ozone_NonSeasonalTrend_Monthly <-
  as.data.frame(Garinger_Monthly_Decomp$time.series[,1:3])

#Add remainder and trend into one column to get data

Garinger_Ozone_NonSeasonalTrend_Monthly <-
  Garinger_Ozone_NonSeasonalTrend_Monthly %>%
  mutate(Nonseasonal = trend + remainder)

#Add date column from the monthly data frame

Garinger_Ozone_NonSeasonalTrend_Monthly$Date <- GaringerOzone.monthly$Date

#Check the data to ensure it has the right columns

#view(Garinger_Ozone_NonSeasonalTrend_Monthly)

#Assign month and year variables

nonseasonal_month <- month(first(Garinger_Ozone_NonSeasonalTrend_Monthly$Date))
nonseasonal_year <- year(first(Garinger_Ozone_NonSeasonalTrend_Monthly$Date))

#Create time series object

GaringerOzone.Nonseasonal.ts <-
  ts(Garinger_Ozone_NonSeasonalTrend_Monthly$Nonseasonal,
     Garinger_Ozone_NonSeasonalTrend_Monthly$Date,
                                   start=c(nonseasonal_year,
                                           nonseasonal_month), frequency=12)

#View time series

#view(GaringerOzone.Nonseasonal.ts)

#16 Run the Mann-Kendall test and view results

Garinger_Ozone_NonSeasonalTrend_Monthly_MKTest <-
  trend::mk.test(Garinger_Ozone_NonSeasonalTrend_Monthly$Nonseasonal)

Garinger_Ozone_NonSeasonalTrend_Monthly_MKTest
```

```
##
##  Mann-Kendall trend test
##
```

```
## data:  Garinger_Ozone_NonSeasonalTrend_Monthly$Nonseasonal
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

```
summary(Garinger_Ozone_NonSeasonalTrend_Monthly)
```

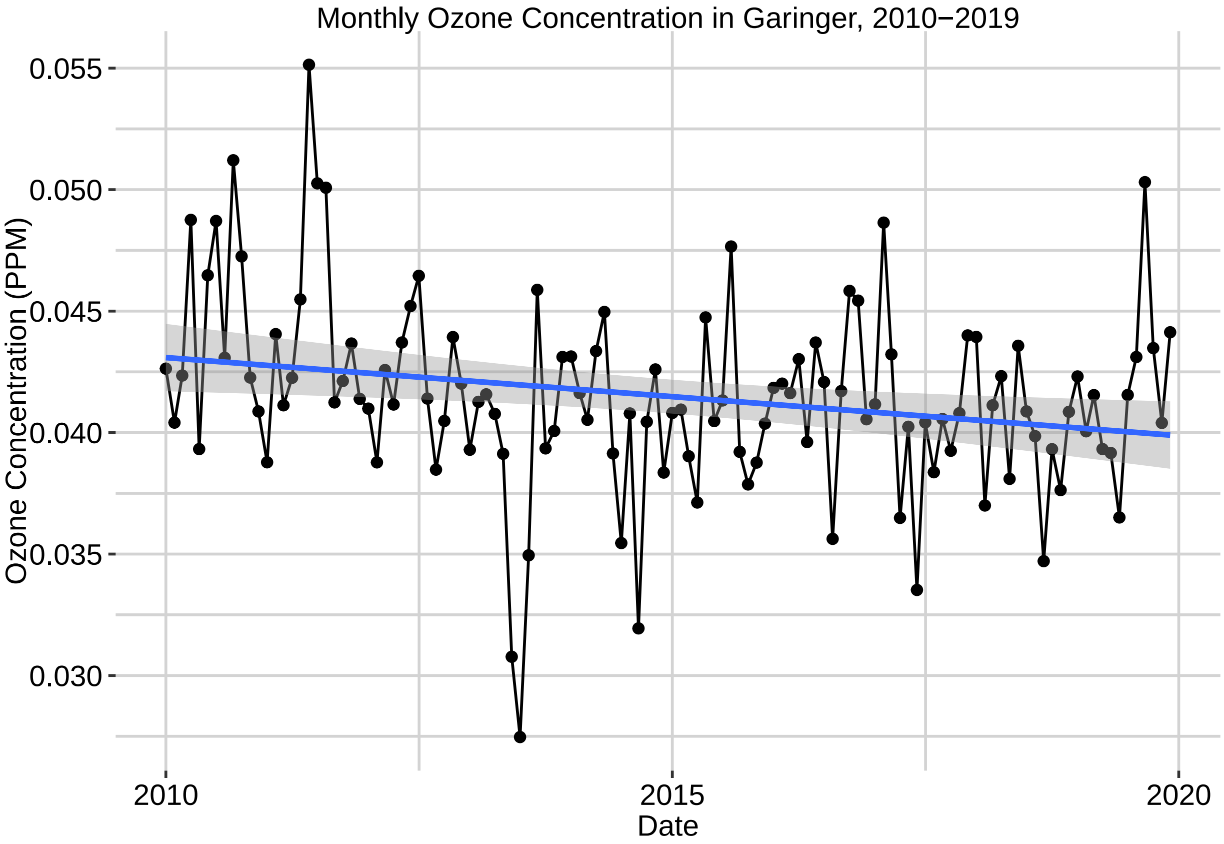
```
##     seasonal              trend           remainder            Nonseasonal
##  Min.   :-0.014935   Min.   :0.03841   Min.   :-1.094e-02   Min.   :0.02747
##  1st Qu.:-0.006976   1st Qu.:0.04013   1st Qu.:-1.987e-03   1st Qu.:0.03932
##  Median : 0.002895   Median :0.04108   Median : 8.770e-05   Median :0.04120
##  Mean   : 0.000000   Mean   :0.04150   Mean   :-1.152e-05   Mean   :0.04149
##  3rd Qu.: 0.006982   3rd Qu.:0.04247   3rd Qu.: 2.049e-03   3rd Qu.:0.04325
##  Max.   : 0.011093   Max.   :0.04504   Max.   : 1.013e-02   Max.   :0.05514
##       Date
##  Min.   :2010-01-01
##  1st Qu.:2012-06-23
##  Median :2014-12-16
##  Mean   :2014-12-16
##  3rd Qu.:2017-06-08
##  Max.   :2019-12-01
```

```
#Plot without seasonality

Garinger_Ozone_NonseasonalTrend_Monthly_Plot <-
ggplot(Garinger_Ozone_NonSeasonalTrend_Monthly,
       aes(x = Date, y = Nonseasonal)) +
  geom_point() +
  geom_line() +  labs(
    title = "Monthly Ozone Concentration in Garinger, 2010-2019",
    x = "Date",
    y = "Ozone Concentration (PPM)") +
  theme(axis.title.y =
            element_text(angle = 90)) +
  geom_smooth( method = lm )

print(Garinger_Ozone_NonseasonalTrend_Monthly_Plot)
```

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



Answer: Without seasonality, the Mann-Kendall test displayed a p-value of 0.00754. The results are still statistically significant, but with a stronger indication of significance than under the seasonal test. This means we would reject the null hypothesis, and accept the alternative hypothesis that there is a correlation between date and ozone concentration over this decade. Both the Non-Seasonal Mann-Kendall and the Seasonal Mann-Kendall tests depict statistical significance in the decrease of ozone concentration measured at Garinger over this decade.

Overall, Garinger experienced a slight decrease in ozone concentration over this decade, with a few points of extremity. Between 2011 and 2012, there is an increase spike in ozone concentration, which could indicate an extreme weather event or man-made issue impacting ozone concentration. Around 2013, there is a decrease spike in the ozone concentration, which could correlate with a cold weather event or other incident reducing ozone levels. By removing seasonality, we can infer that the variability in this graph's ozone concentrations must be due to factors other than the season.