Assignment 8: Time Series Analysis

Keanu Valibia

Spring 2024

Contents

Overview 1
Directions 1
Set up 2
Wrangle 3
Visualize 5
Time Series Analysis 6

List of Figures

1 Ozone Concentrations Over Time 5
2 Decomposition Plots 7
3 Decomposition Plots 8
4 Mean Monthly Ozone Concentrations 9
5 Component Trend Analysis 10
6 Component Seasonal Analysis 11

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

```
# install.packages('zoo')
# install.packages('trend')
# install.packages('Kendall')
library(tidyverse)
library(lubridate)
library(zoo)
library(here)
library(trend)
library(Kendall)

here()

myTheme <- theme_classic(base_size = 11) + theme(axis.text = element_text(color = "black"),
    axis.line = element_line(color = "#320E3B"),
    panel.background = element_rect(fill = "#EDE6E3"),
    panel.grid.major = element_line(color = "#36382E",
        linetype = "dotted"), plot.title = element_text(size = 15),
    axis.title.x = element_text(size = 13), axis.title.y = element_text(size = 13),
    legend.position = "right")

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.

```
# 1

GaringerOzone <- list.files(path = "~/R/R Projects/EDA_Spring2024/Data/Raw/Ozone_TimeSeries",
    pattern = "*.csv", full.names = TRUE) %>%
    lapply(read.csv) %>%
    bind_rows

glimpse(GaringerOzone)
```

```
## Rows: 3,589
## Columns: 20
## $ Date                                  <chr> "01/01/2010", "01/02/2010", "01/0~
## $ Source                                <chr> "AQS", "AQS", "AQS", "AQS", "AQS"~
## $ Site.ID                               <int> 371190041, 371190041, 371190041, ~
## $ POC                                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ UNITS                                 <chr> "ppm", "ppm", "ppm", "ppm", "ppm"~
```

```
## $ DAILY_AQI_VALUE                       <int> 29, 31, 32, 29, 25, 31, 32, 30, 3~
## $ Site.Name                             <chr> "Garinger High School", "Garinger~
## $ DAILY_OBS_COUNT                       <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                      <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                    <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                    <chr> "Ozone", "Ozone", "Ozone", "Ozone~
## $ CBSA_CODE                             <int> 16740, 16740, 16740, 16740, 16740~
## $ CBSA_NAME                             <chr> "Charlotte-Concord-Gastonia, NC-S~
## $ STATE_CODE                            <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                                 <chr> "North Carolina", "North Carolina~
## $ COUNTY_CODE                           <int> 119, 119, 119, 119, 119, 119, 119~
## $ COUNTY                                <chr> "Mecklenburg", "Mecklenburg", "Me~
## $ SITE_LATITUDE                         <dbl> 35.2401, 35.2401, 35.2401, 35.240~
## $ SITE_LONGITUDE                        <dbl> -80.78568, -80.78568, -80.78568, ~
```

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")
glimpse(GaringerOzone)
```

```
## Rows: 3,589
## Columns: 20
## $ Date                                  <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Source                                <chr> "AQS", "AQS", "AQS", "AQS", "AQS"~
## $ Site.ID                               <int> 371190041, 371190041, 371190041, ~
## $ POC                                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ UNITS                                 <chr> "ppm", "ppm", "ppm", "ppm", "ppm"~
## $ DAILY_AQI_VALUE                       <int> 29, 31, 32, 29, 25, 31, 32, 30, 3~
## $ Site.Name                             <chr> "Garinger High School", "Garinger~
## $ DAILY_OBS_COUNT                       <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                      <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                    <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                    <chr> "Ozone", "Ozone", "Ozone", "Ozone~
## $ CBSA_CODE                             <int> 16740, 16740, 16740, 16740, 16740~
## $ CBSA_NAME                             <chr> "Charlotte-Concord-Gastonia, NC-S~
## $ STATE_CODE                            <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
```

```
## $ STATE                                  <chr> "North Carolina", "North Carolina~
## $ COUNTY_CODE                            <int> 119, 119, 119, 119, 119, 119, 119~
## $ COUNTY                                 <chr> "Mecklenburg", "Mecklenburg", "Me~
## $ SITE_LATITUDE                          <dbl> 35.2401, 35.2401, 35.2401, 35.240~
## $ SITE_LONGITUDE                         <dbl> -80.78568, -80.78568, -80.78568, ~
```

```
# 4

GaringerOzone.slim <- select(GaringerOzone, Date,
    Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
glimpse(GaringerOzone.slim)
```

```
## Rows: 3,589
## Columns: 3
## $ Date                                   <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ DAILY_AQI_VALUE                        <int> 29, 31, 32, 29, 25, 31, 32, 30, 3~
```

```
# 5

Days <- as.data.frame(seq(as.Date("2010-01-01"),
    as.Date("2019-12-31"), "day"))
colnames(Days)[1] <- "Date"
glimpse(Days)
```

```
## Rows: 3,652
## Columns: 1
## $ Date <date> 2010-01-01, 2010-01-02, 2010-01-03, 2010-01-04, 2010-01-05, 2010~
```

```
# 6

GaringerOzone <- left_join(Days, GaringerOzone,
    by = "Date")
glimpse(GaringerOzone)
```

```
## Rows: 3,652
## Columns: 20
## $ Date                                   <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Source                                 <chr> "AQS", "AQS", "AQS", "AQS", "AQS"~
## $ Site.ID                                <int> 371190041, 371190041, 371190041, ~
## $ POC                                    <int> 1, 1, 1, 1, 1, NA, 1, 1, 1, 1, 1,~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ UNITS                                  <chr> "ppm", "ppm", "ppm", "ppm", "ppm"~
## $ DAILY_AQI_VALUE                        <int> 29, 31, 32, 29, 25, NA, 31, 32, 3~
## $ Site.Name                              <chr> "Garinger High School", "Garinger~
## $ DAILY_OBS_COUNT                        <int> 17, 17, 17, 17, 17, NA, 17, 17, 1~
## $ PERCENT_COMPLETE                       <dbl> 100, 100, 100, 100, 100, NA, 100,~
## $ AQS_PARAMETER_CODE                     <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                     <chr> "Ozone", "Ozone", "Ozone", "Ozone~
## $ CBSA_CODE                              <int> 16740, 16740, 16740, 16740, 16740~
## $ CBSA_NAME                              <chr> "Charlotte-Concord-Gastonia, NC-S~
## $ STATE_CODE                             <int> 37, 37, 37, 37, 37, NA, 37, 37, 3~
```

```
## $ STATE                                <chr> "North Carolina", "North Carolina~
## $ COUNTY_CODE                          <int> 119, 119, 119, 119, 119, NA, 119,~
## $ COUNTY                               <chr> "Mecklenburg", "Mecklenburg", "Me~
## $ SITE_LATITUDE                        <dbl> 35.2401, 35.2401, 35.2401, 35.240~
## $ SITE_LONGITUDE                       <dbl> -80.78568, -80.78568, -80.78568, ~
```

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

GaringerOzone.plot <- ggplot(GaringerOzone, aes(x = Date,
    y = Daily.Max.8.hour.Ozone.Concentration,
    )) + geom_line() + labs(title = "Ozone Concentrations Over Time",
    caption = "Source: EPA Air Database (Garinger High School, NC)") +
    xlab("Date") + ylab("Daily 8-Hour Ozone Concentration (PPM)") +
    scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
    geom_smooth(method = "lm", color = "#320E3B")

GaringerOzone.plot
```

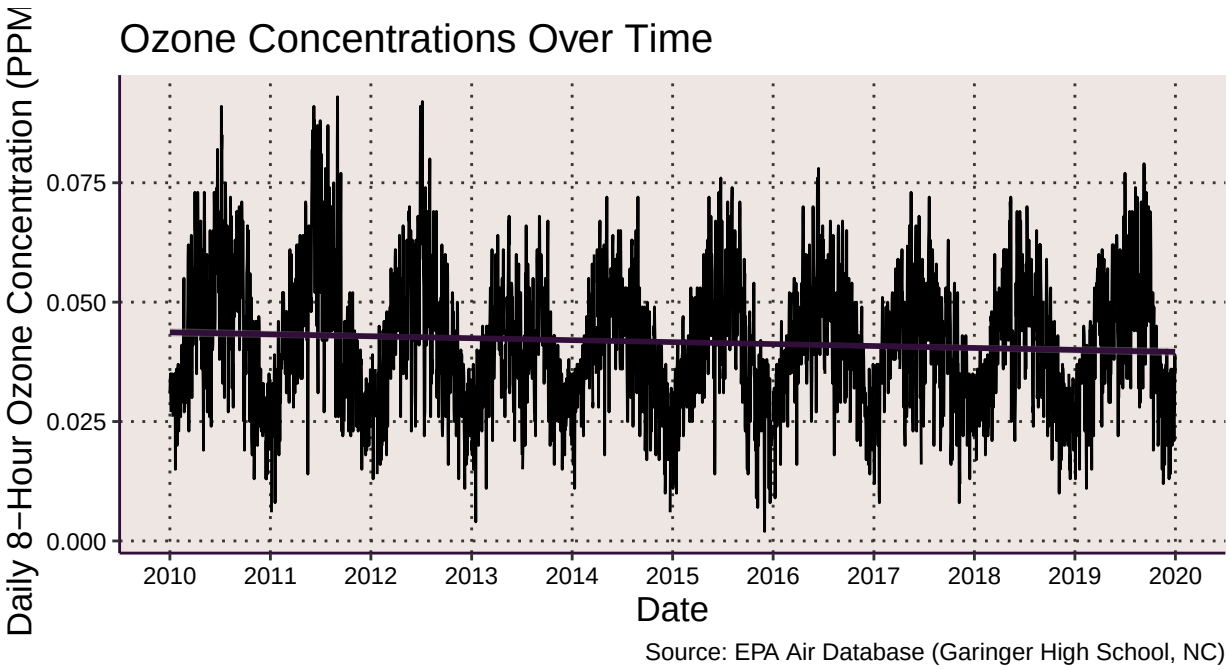

Figure 1: Ozone Concentrations Over Time

Answer: Linear trend seems to indicate a very slight decrease in 8-hour ozone concentrations 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

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

head(GaringerOzone$ozone.clean)
```

```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

Answer: Spline interpolation would be innaproppriate considering that our data seems to be moving most in a linear fashion, rather than a polynomial function. Piecewise constant would be innappropriate since it assumes a constant value based on nearest neighbors, which contradicts the constant movement of our data.

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(Year = year(ymd(Date))) %>%
    mutate(Month = month(ymd(Date))) %>%
    group_by(Year, Month) %>%
    summarise(mean_ozone = mean(ozone.clean))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
    mutate(DateGroup = make_date(Year, 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.

```
# 10

f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))
```

```
GaringerOzone.daily.ts <- ts(GaringerOzone$ozone.clean,
    start = c(f_year, f_month), frequency = 365)

f_month2 <- first(GaringerOzone.monthly$Month)
f_year2 <- first(GaringerOzone.monthly$Year)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
    start = c(f_year2, f_month2), 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)
```

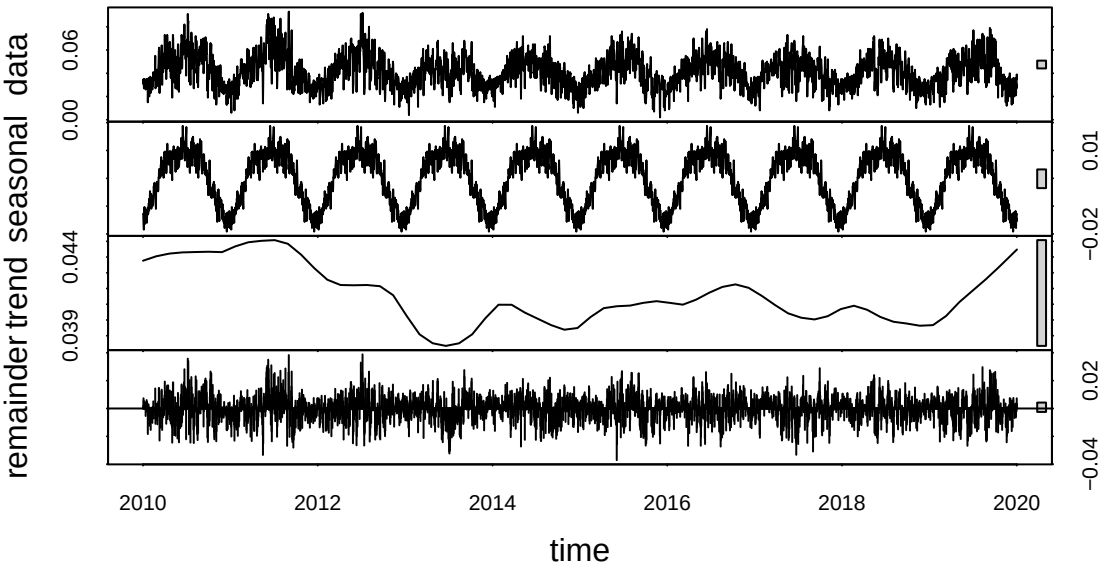

Figure 2: Decomposition Plots

```
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?

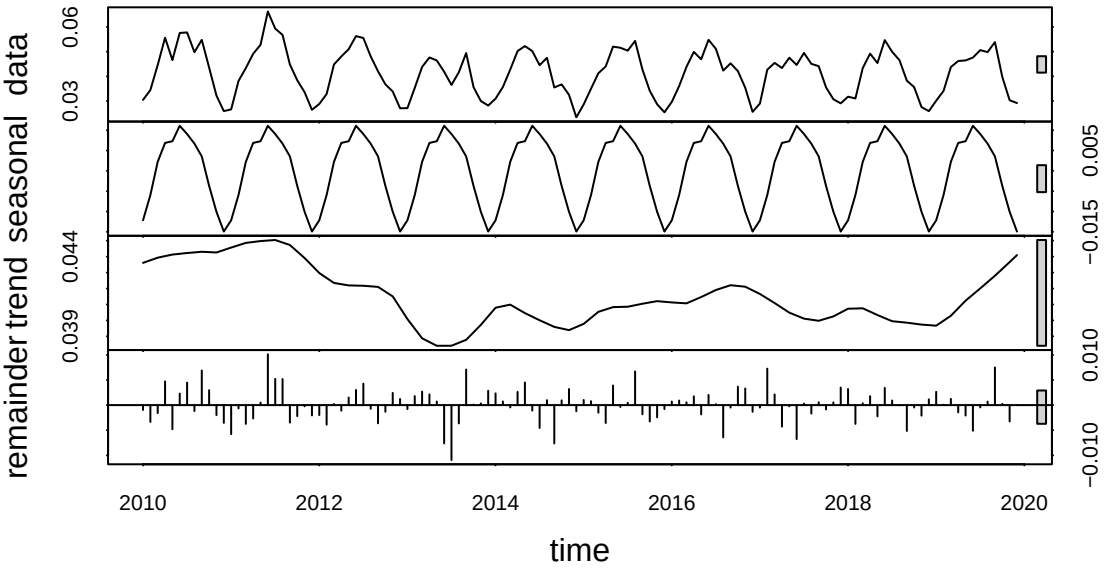

Figure 3: Decomposition Plots

```
# 12

(GaringerOzone.monthly.ts)
```

```
##             Jan        Feb        Mar        Apr        May        Jun
## 2010 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
## 2011 0.02661290 0.03810714 0.04335484 0.04913333 0.05277419 0.06623333
## 2012 0.02882258 0.03282759 0.04480645 0.04803333 0.05100000 0.05630000
## 2013 0.02712903 0.03532143 0.04380645 0.04765000 0.04641935 0.04186667
## 2014 0.03096774 0.03567857 0.04275806 0.05023333 0.05225806 0.05023333
## 2015 0.02864516 0.03500000 0.04125806 0.04400000 0.05203226 0.05156667
## 2016 0.02967742 0.03606897 0.04385484 0.04990000 0.04690323 0.05480000
## 2017 0.02900000 0.04269643 0.04545161 0.04336667 0.04753226 0.04461667
## 2018 0.03177419 0.03105357 0.04335484 0.04920000 0.04538710 0.05466667
## 2019 0.03014516 0.03410714 0.04377419 0.04620000 0.04645161 0.04760000
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2010 0.05777419 0.04977419 0.05476667 0.04354839 0.03220000 0.02593548
## 2011 0.05932258 0.05677419 0.04480000 0.03841935 0.03360000 0.02645161
## 2012 0.05551613 0.04809677 0.04203333 0.03677419 0.03386667 0.02708065
## 2013 0.03653226 0.04164516 0.04943333 0.03564516 0.03000000 0.02817742
## 2014 0.04451613 0.04748387 0.03550000 0.03674194 0.03253333 0.02341935
## 2015 0.05038710 0.05435484 0.04276667 0.03416129 0.02870000 0.02543548
## 2016 0.05114516 0.04232258 0.04526667 0.04212903 0.03536667 0.02561290
## 2017 0.04948387 0.04506452 0.04411667 0.03554839 0.03073333 0.02906452
## 2018 0.04993548 0.04654839 0.03826667 0.03561290 0.02756667 0.02591935
## 2019 0.05061290 0.04980645 0.05386667 0.03977419 0.03033333 0.02919355
```

Answer: The data includes seasonality, and we are examining a fully-filled dataset we cleaned

using linear interpolation (i.e. there is no missing 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

GaringerOzone.plot2 <- ggplot(GaringerOzone.monthly,
    aes(x = DateGroup, y = mean_ozone)) + geom_point() +
    geom_line() + labs(title = "Mean Monthly Ozone Concentrations",
    caption = "Source: Source: EPA Air Database (Garinger High School, NC)") +
    xlab("Date") + scale_x_date(date_breaks = "1 year",
    date_labels = "%Y") + ylab("Monthly Mean Ozone Concentration (PPM)")

GaringerOzone.plot2
```

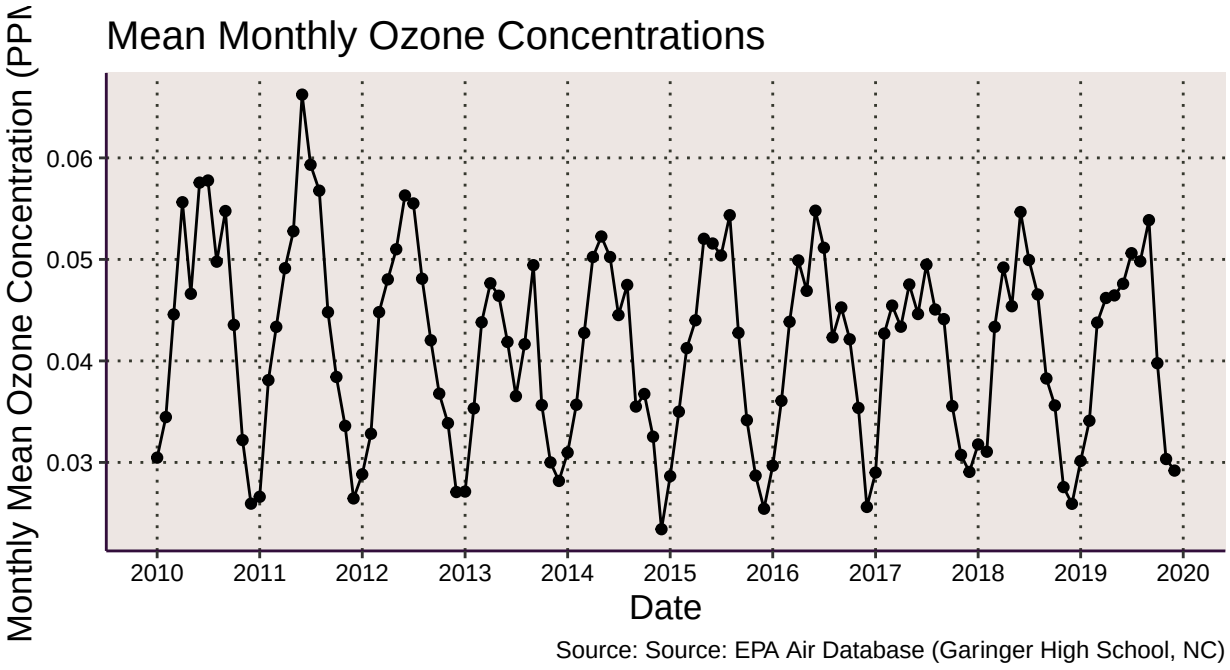

Figure 4: Mean Monthly Ozone Concentrations

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: Our study question asks whether ozone concentrations have changed over time. Graphical analysis seems to indicate only a slight negative trend in ozone concentrations over the past decade. However, the Seasonal MannKendall analysis indicates a very slight negative monotonic trend over this 10-year time period. This result is statistically significant at a 95% Confidence Level. There is enough reason to reject the null hypothesis (i.e. there is reason to reject the hypothesis that there are no changes in ozone concentrations over time). (tau = -0.143, 2-sided p-value = 0.046724).

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

Garinger.components <- mutate(Garinger.components,
    Observed = GaringerOzone.monthly$mean_ozone,
    Date = GaringerOzone.monthly$DateGroup)

Garinger.observed.plot <- ggplot(Garinger.components) +
    geom_line(aes(y = Observed, x = Date, color = "Observed")) +
    geom_line(aes(y = trend, x = Date, color = "Trend")) +
    geom_hline(yintercept = 0, lty = 2) + labs(title = "Mean Monthly Ozone Concentration Trends",
    caption = "Source: Source: EPA Air Database (Garinger High School, NC)",
    color = NULL) + xlab("Date") + scale_x_date(date_breaks = "1 year",
    date_labels = "%Y") + ylab("Observed Ozone Concentration (PPM)") +
    scale_color_manual(values = c(Observed = "black",
        Trend = "#320E3B"), labels = c(Observed = "Observed",
        Trend = "Trend"), limits = c("Observed",
        "Trend"))

Garinger.observed.plot
```

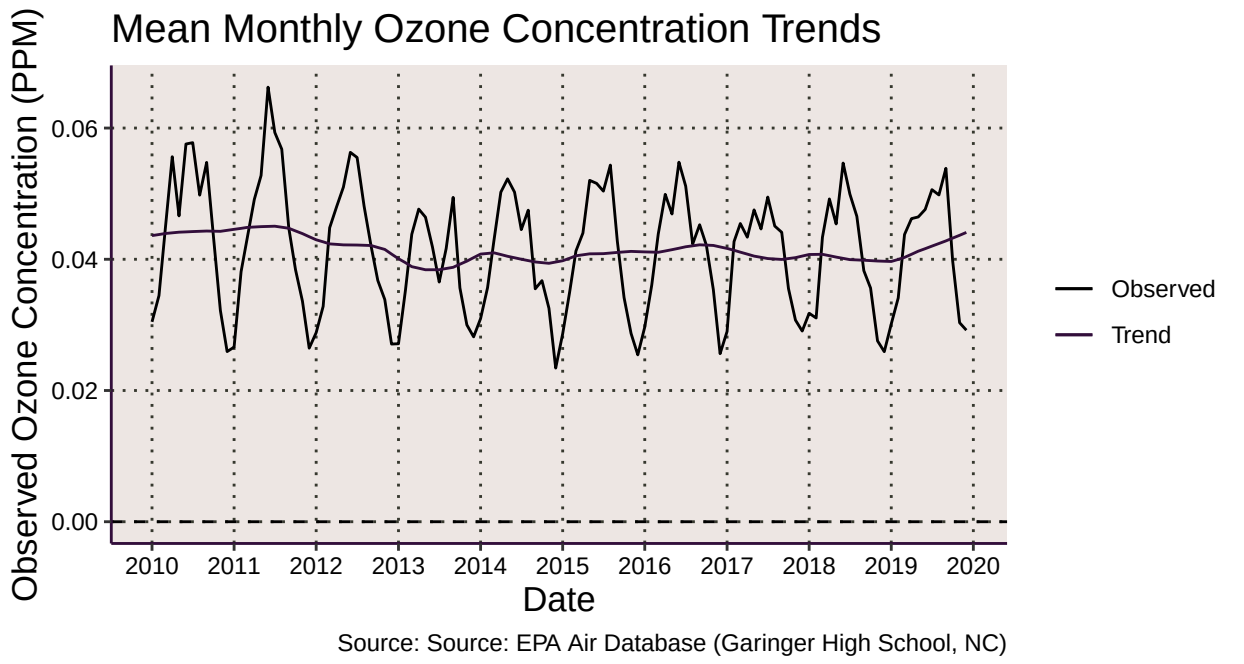

Figure 5: Component Trend Analysis

```
Garinger.seasonal.plot <- ggplot(Garinger.components) +
    geom_line(aes(y = Observed, x = Date, color = "Observed")) +
    geom_line(aes(y = seasonal, x = Date, color = "Seasonal")) +
    geom_hline(yintercept = 0, lty = 2) + labs(title = "Seasonal & Mean Monthly Ozone Concentration",
    caption = "Source: Source: EPA Air Database (Garinger High School, NC)",
    color = NULL) + xlab("Date") + scale_x_date(date_breaks = "1 year",
    date_labels = "%Y") + ylab("Ozone Concentration (PPM)") +
    scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
    scale_color_manual(values = c(Observed = "black",
        Seasonal = "#320E3B"), labels = c(Observed = "Observed",
        Seasonal = "Seasonal"), limits = c("Observed",
        "Seasonal"))

Garinger.seasonal.plot
```

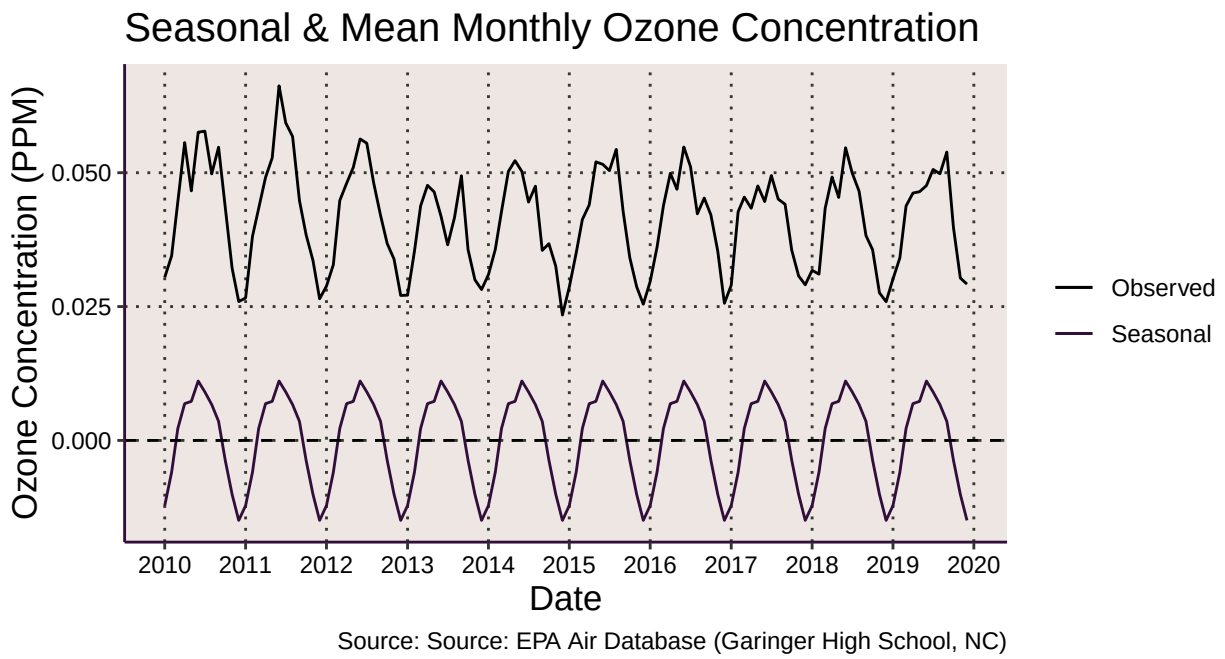

Figure 6: Component Seasonal Analysis

```
# 16

GaringerOzone.monthly.nonseasonal.ts <- ts(Garinger.components$trend,
    start = c(f_year2, f_month2), frequency = 12)

SeasonalMannKendall(GaringerOzone.monthly.nonseasonal.ts)
```

```
## tau = -0.304, 2-sided pvalue =2.291e-05
```

Answer: Analyzing non-seasonal data indicates a more moderate decline in monotonic trend, while still maintaining statistical significance at both the 95% and 99% Confidence Levels. These results are more pronounced than the composite analysis conducted earlier, which included seasonality. (Tau = -0.304, 2-sided p-value = 0.00002291)