Assignment 8: Time Series Analysis

Iman Byndloss

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

```
#1
# Used getwd() to check the working directory.
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
# Used library() to load relevant packages.
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(zoo)
```

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

```
library(trend)

library(ggthemes)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
# COPIED FROM A05: Used theme_base() + theme() to establish a custom theme.
custom_grid <- theme_base() +
  theme(
    panel.background = element_rect(fill="white"),
    panel.grid.major = element_line(color="gray", linewidth = 0.4),
    panel.grid.minor = element_line(color="gray", linewidth = 0.2),
    # Set a grey background with white grid lines
    plot.title = element_text(size=10, face="bold",hjust=0.5),
    # Set text size, emboldened, and centered for plot title
    axis.title = element_text(size=8),
    # Set text size for axis title
    axis.ticks = element_line(color="black"),
    axis.ticks.length=unit(0.15,"cm"),
    # Set length of axis ticks
    legend.box.background = element_rect(color="black", size=0.5),
    # Set a black border around legend
    strip.text = element_text(face="bold", size=8),
    # Set the facet labels to bold small text
    legend.position = "right"
    # Set legend position to right of graphs
    )
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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
#2
# Used read.csv() and here() to import specified raw data files.
G2010 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)

G2011 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)

G2012 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)

G2013 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)

G2014 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)

G2015 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)

G2016 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)

G2017 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)

G2018 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)

G2019 <- read.csv(here(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)

# Used rbind() to combine datasets for each year into one large data frame.
GaringerOzone <- rbind(G2010,
                       G2011,
                       G2012,
                       G2013,
                       G2014,
                       G2015,
                       G2016,
                       G2017,
                       G2018,
                       G2019)
```

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
# Used as.Date() to establish sampledate as a date with the format %m/%d/%Y.
# The four digit year corresponds to "%Y", not "%y".
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format="%m/%d/%Y")

#4
# Used select() to choose only relevant columns for the data frame.
GaringerOzone <- select(GaringerOzone,
                        Date,
                        Daily.Max.8.hour.Ozone.Concentration,
                        DAILY_AQI_VALUE)

#5
# Used as.data.frame() & seq() to create a new data frame for select dates.
Days <- as.data.frame(seq(from=as.Date("01/01/2010", format="%m/%d/%Y"),
                          to=as.Date("12/31/2019", format="%m/%d/%Y"),
                          by="day"))

# Used colnames() & c() to rename the column.
colnames(Days) <- c("Date")

#6
# Combined two data frames with left_join().
GaringerOzone <- left_join(Days, GaringerOzone, by="Date")
```

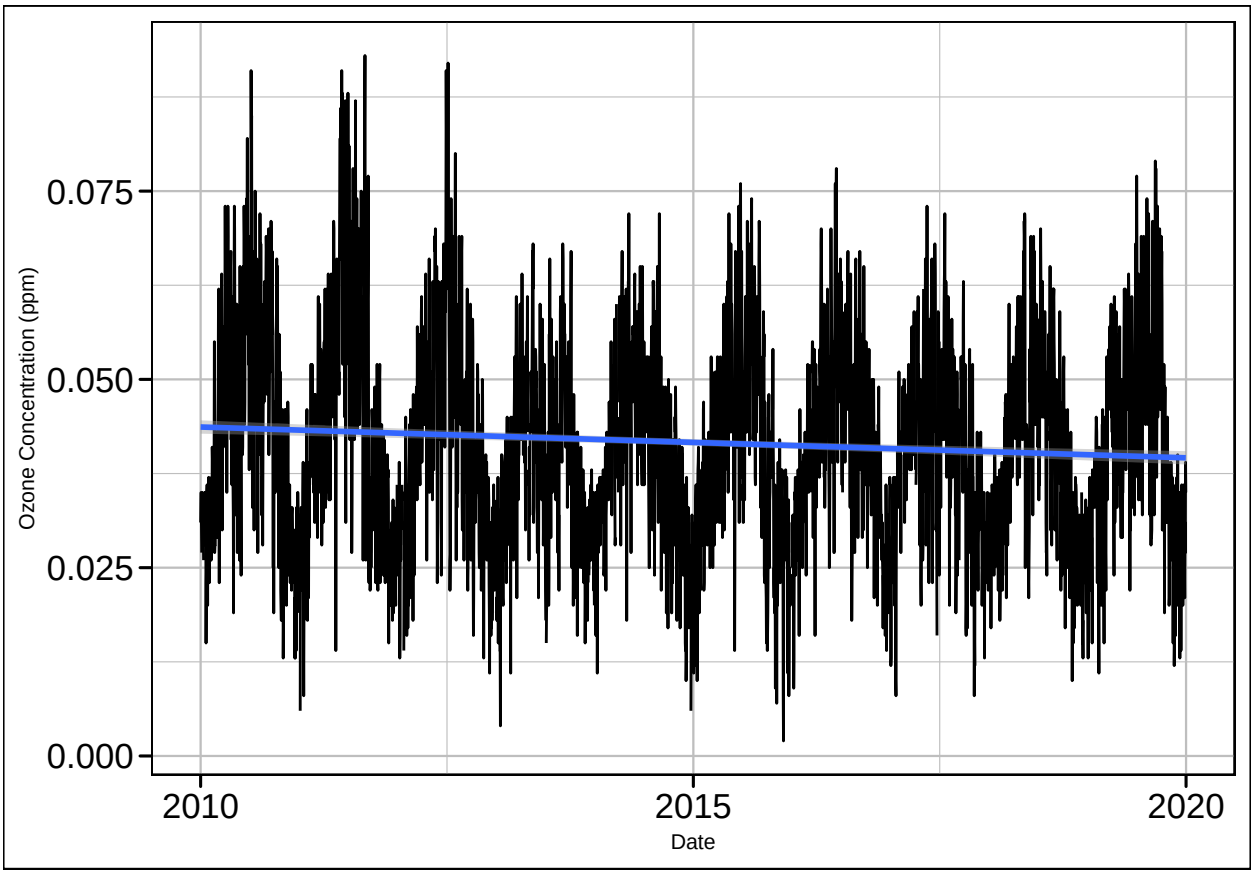
Visualize

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
# Used ggplot() to create a line plot of ozone concentrations over time.
ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method="lm") +
  labs(x="Date",y="Ozone Concentration (ppm)") +
  custom_grid
```

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

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



Answer: Based on the plot, there is a slight decrease in ozone concentrations (ppm) 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
# Added new column with no missing observations through linear interpolation
GaringerOzoneClean <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzoneClean$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: There are three types of interpolation: 1) piecewise constant, 2) linear, and 3) spline. Piecewise constant interpolation assumes any missing data are equal to the closest measurement to that date. Linear interpolation assumes any missing data are between the previous and next measurements, drawing a straight line between known points. Lastly, spline interpolation is similar to linear interpolation except it uses a quadratic function instead of a straight line. In the case of this research, linear interpolation is best due to the fact that the relationship is linear and displays gradual daily changes. Notably, spline interpolation does not display linear relationships and piecewise constant interpolation is not ideal for more gradual changes.

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)

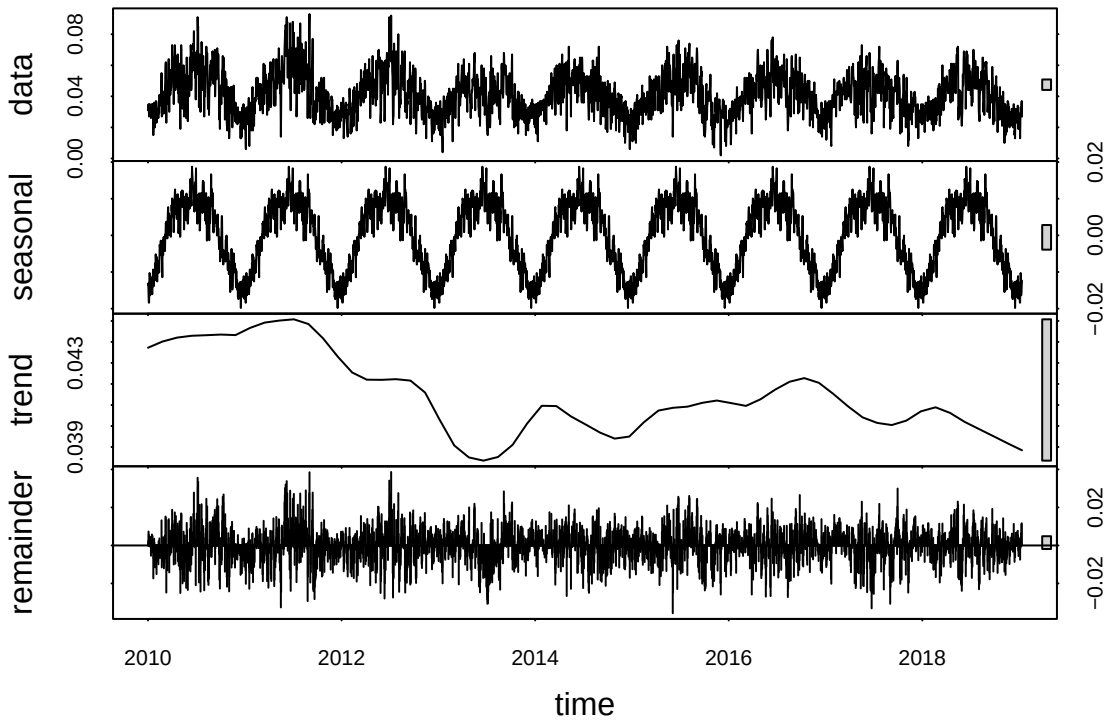
```
#9
# Created a new data frame with monthly mean ozone concentrations.
# Also, added a new date column with each month-year combination.
GaringerOzone.monthly <- GaringerOzoneClean %>%
  mutate(Month=month(Date), Year=year(Date)) %>%
  group_by(Year, Month) %>% # group by year then by month
  summarise(meanO3 = mean(Daily.Max.8.hour.Ozone.Concentration),
            .groups="keep") %>%
  mutate(Date = format(make_date(Year,Month,1), "%m/%d/%Y"))
```

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.

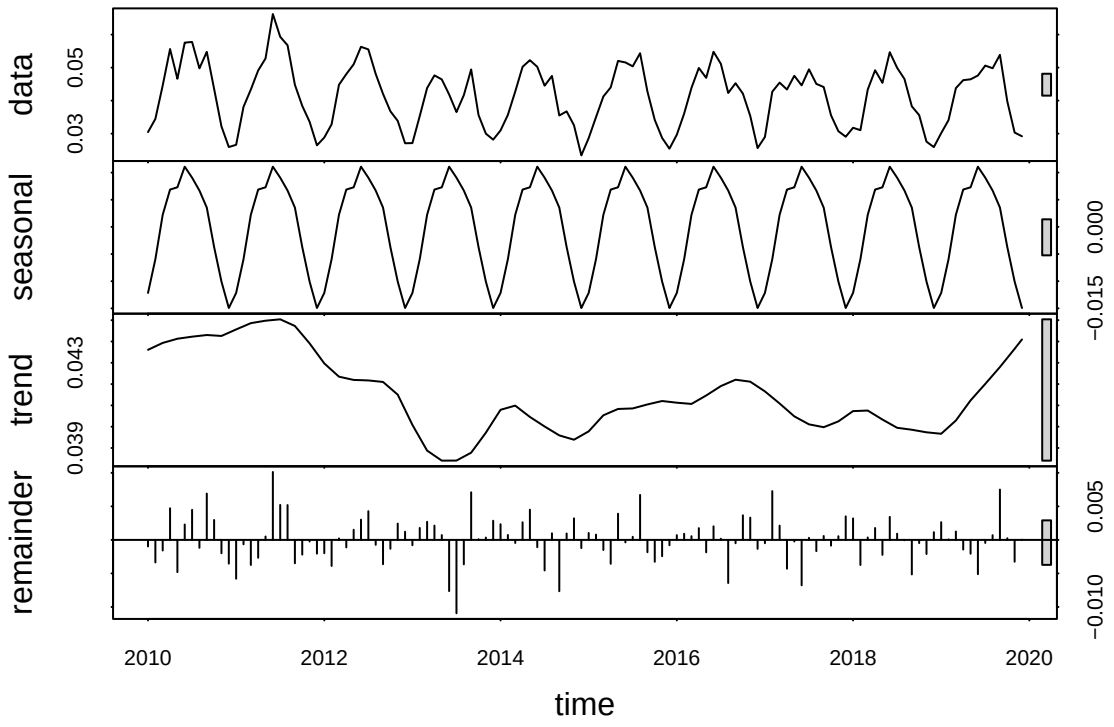
```
#10
# Generated two time series, one daily(freq=365) and the other monthly(freq=12)
GaringerOzone.daily.ts <- ts(
  GaringerOzoneClean$Daily.Max.8.hour.Ozone.Concentration,
                   start=c(2010,1),
                   end=c(2019,12),
                   frequency=365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanO3,
                   start=c(2010,1),
                   end=c(2019,12),
                   frequency=12)
```

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

```
#11
# Used stl() with s.window="periodic" to decompose the two time series.
# Used plot() to plot the results of the decomposition.
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```
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?

```
#12
# Ran the seasonal Mann-Kendall test with Kendall::SeasonalMannKendall().
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(
  GaringerOzone.monthly.ts)

# Inspected the results through printing and with summary().
GaringerOzone.monthly.trend
```

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

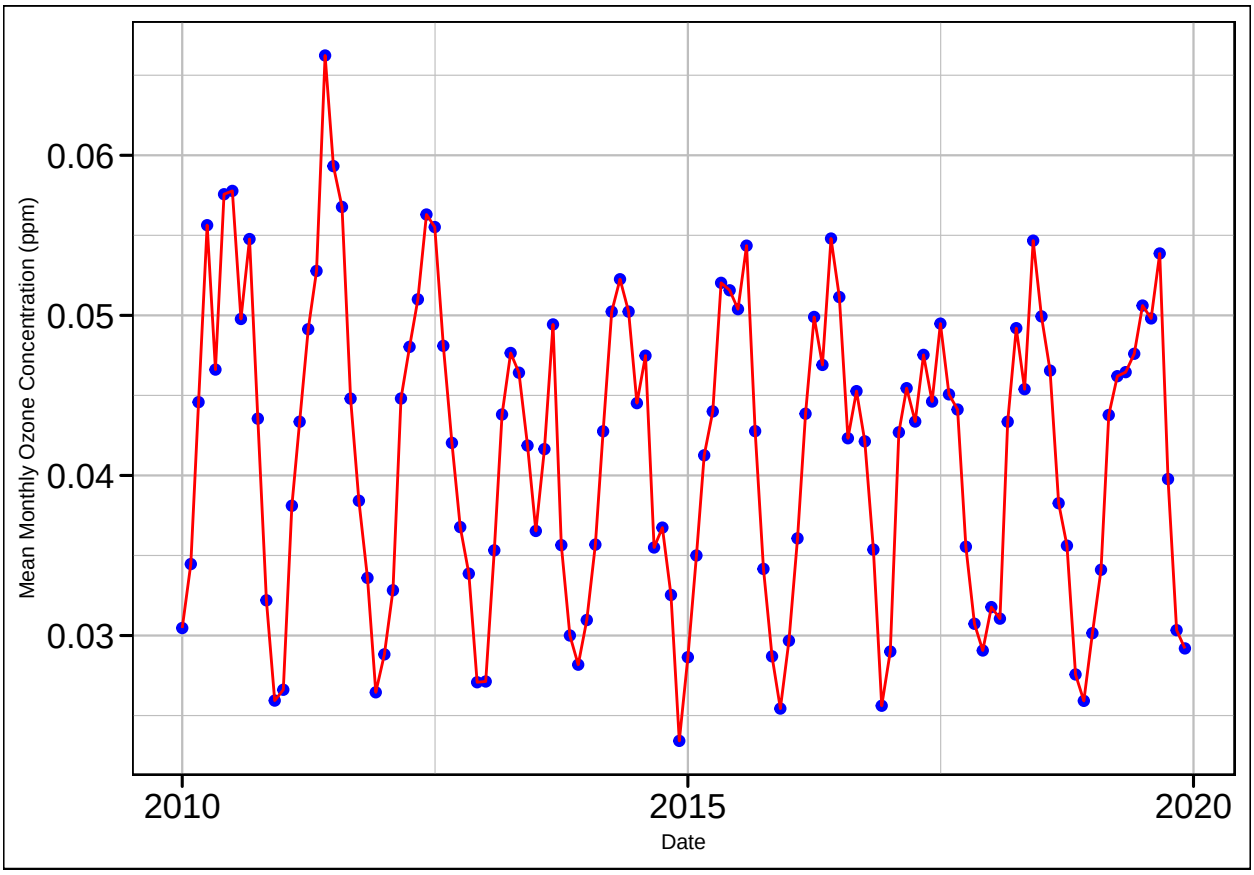
```
summary(GaringerOzone.monthly.trend)
```

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

Answer: For the monthly Ozone series, "seasonal" has one of the smallest bars on the plot, suggesting it is one of the most important factors considered for this trend. To better understand the seasonality of this monthly Ozone series, a seasonal Mann-Kendall test can be conducted. Notably, the other tests for monotonic trend analysis (linear regression, Mann-Kendall, and Spearman Rho) do not account for 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
# Ensured the date column of the monthly data frame was recognized as a date.
GaringerOzone.monthly$Date <- as.Date(GaringerOzone.monthly$Date,
                                      format="%m/%d/%Y")

# Used ggplot() to create a plot of mean monthly ozone concentrations over time.
ggplot(GaringerOzone.monthly, aes(x=Date, y=meanO3)) +
  geom_point(color="blue") +
  geom_line(color="red") +
  labs(x="Date",y="Mean Monthly Ozone Concentration (ppm)") +
  custom_grid
```



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: As a reminder, the research question was as follows: Have ozone concentrations changed over the 2010s at this station? Based on the results, there is a signficant change in ozone concentration over the 2010s at this station, with number slightly decreasing as time passes (tau=-0.143,2-sided pvalue=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
# Extracted the seasonal component.
GaringerOzone.monthly.seasonal <-
  GaringerOzone.monthly.decomp$time.series[,1]

# Subtracted the seasonal component from the monthly time series.
GaringerOzone.monthly.nonseasonal <-
  GaringerOzone.monthly.ts - GaringerOzone.monthly.seasonal

#16
# Ran the Mann-Kendall test on data frame without seasonal.
GaringerOzone.monthly.nonseasonal.trend <- Kendall::MannKendall(
  GaringerOzone.monthly.nonseasonal)

# Inspected the results through printing and with summary().
GaringerOzone.monthly.nonseasonal.trend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(GaringerOzone.monthly.nonseasonal.trend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: In the seasonal Mann-Kendall test for the seasonal Ozone monthly series, there was a statistically significant weak negative trend in ozone concentrations over time (tau=-0.143,2-sided pvalue=0.046724). In comparison, the Mann-Kendall test for the non-seasonal Ozone monthly series also indicated a weak negative trend, but that trend was slightly stronger and more statistically significant (tau=-0.165, 2-sided p-value=0.0075402). With the removal of seasonality increasing the significance of the displayed trend, there is confirmation that ozone concentrations have been decreasing over the years despite any seasonal variations.