Assignment 7: Time Series Analysis

Katie Krejsa

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 Tuesday, March 16 at 11:59 pm.

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
air_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors =
air_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors =
air_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors =
air_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors =
air_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors =
air_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors =
air_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors =
air_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors =
air_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors =
air_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors =

GaringerOzone <- rbind(air_2010, air_2011, air_2012, air_2013, air_2014, air_2015, air_2016, air_2017, a
```

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

# 4
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
GaringerOzone <- dplyr::select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALU

# 5
Days <- seq.Date(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 days")
Days <- as.data.frame(Days)
colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone)
```

```
## Joining, by = "Date"
```

Visualize

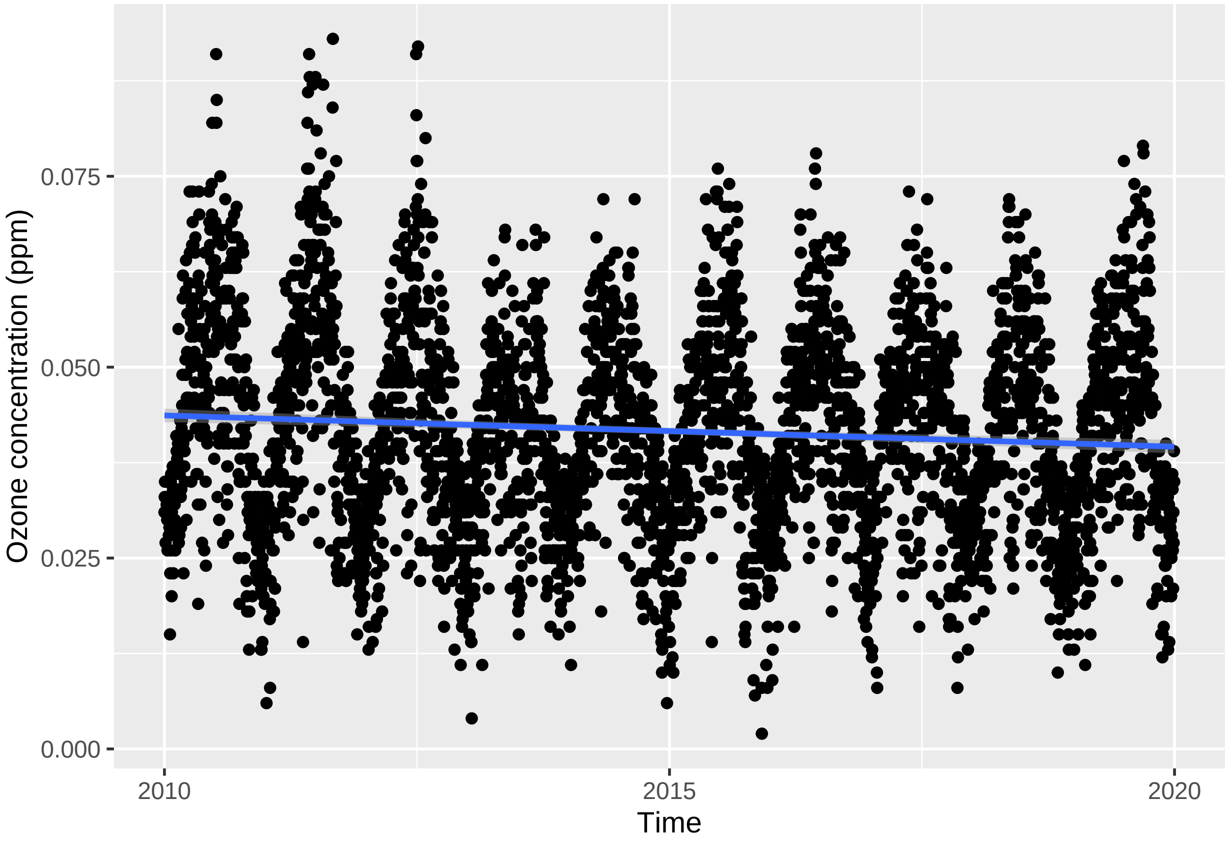
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) + geom_point() +
  geom_smooth(method = lm) +
  ylab("Ozone concentration (ppm)") +
  xlab("Time")
```

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

```
## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 63 rows containing missing values (geom_point).
```



Answer: Yes, my plot suggests a negative trend in ozone concentration over time.

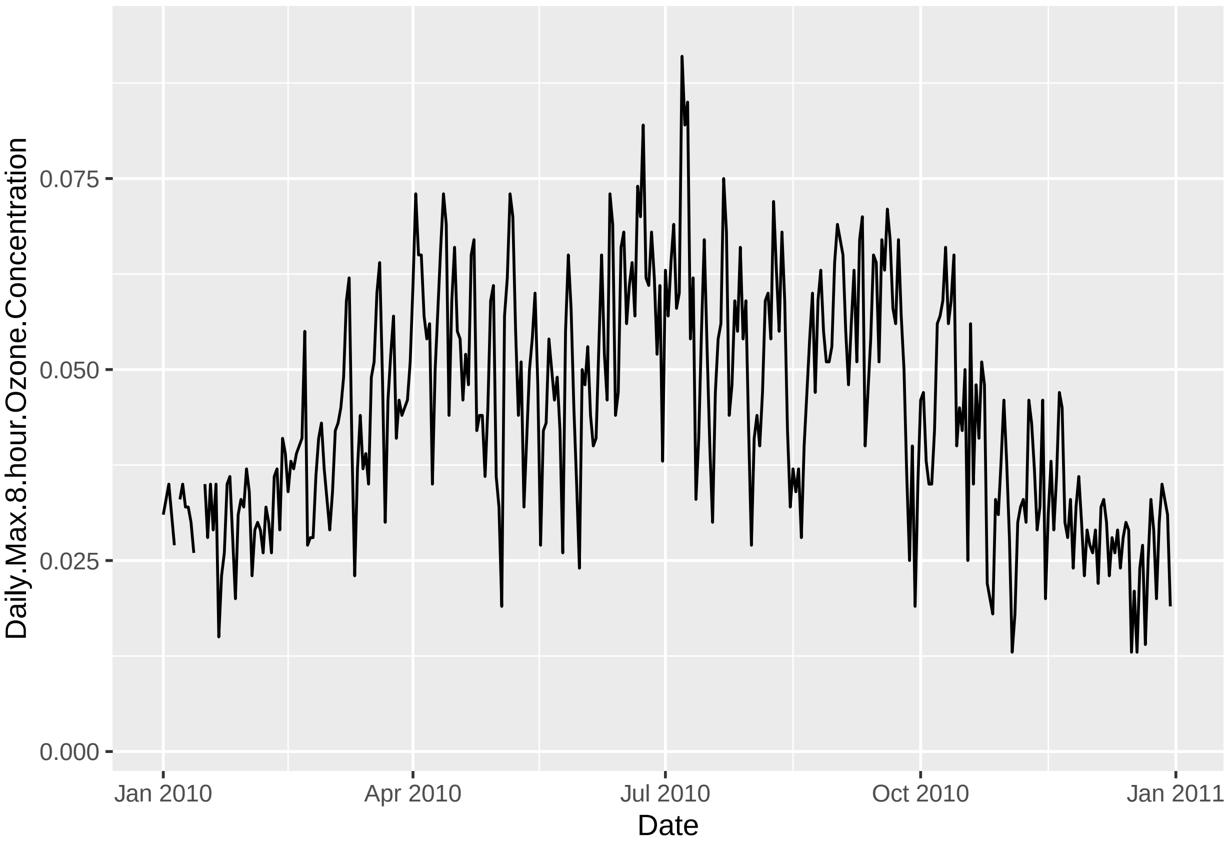
Time Series Analysis

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) + geom_line() +
  scale_x_date(limits = c(as.Date("2010-01-01"), as.Date("2010-12-31")))
```

```
## Warning: Removed 3288 row(s) containing missing values (geom_path).
```



```r
# Can see some missing data

# summarizing the Ozone concentration column and you can see there there are 63 NA's
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```r
# na.approx() function can be used to interpolate missing observations using either linear interpolatio
GaringerOzone.complete <-
  GaringerOzone %>%
  mutate( Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )

summary(GaringerOzone.complete$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
```

```
#The NAs are gone
```

Answer: We did not use a piecewise constant because we wanted to "connect the dots" between concentrations values at two different times, and there are small enough gaps in the data that a linear interpolation can be assumed. We did not use spline interpolation because the gaps of missing data were relatively small and spline interpolation is not necessary for that.

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.complete %>%
  mutate(month = lubridate::month(Date)) %>%
  mutate(year = lubridate::year(Date)) %>%
  group_by(month, year) %>%
  summarise(mean.ozone.conc = mean(Daily.Max.8.hour.Ozone.Concentration))
```

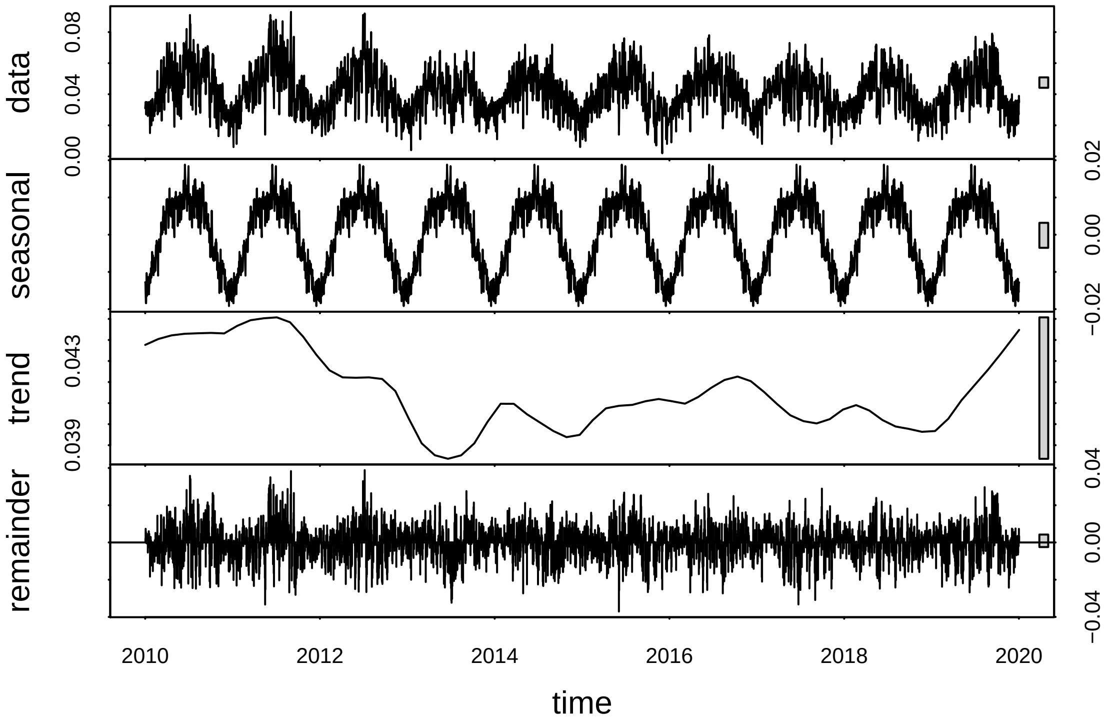
```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argument.
```

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.

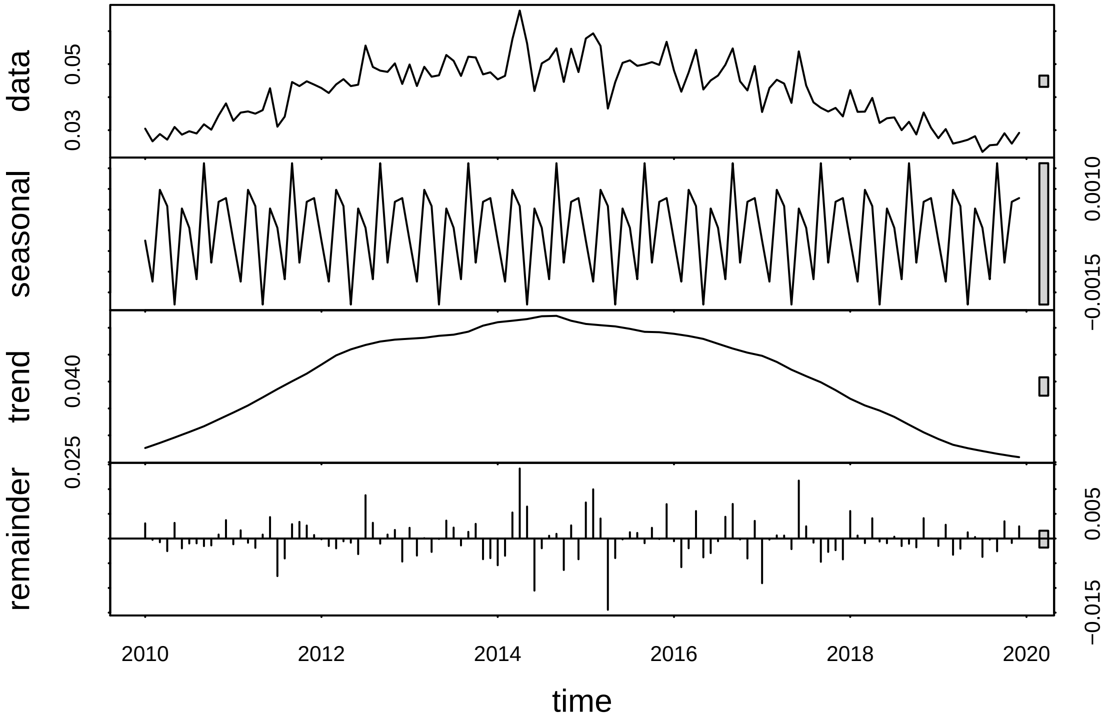
```
#10

GaringerOzone.daily.ts <- ts(GaringerOzone.complete$Daily.Max.8.hour.Ozone.Concentration, start = c(201

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.ozone.conc, start = c(2010,1), frequency = 12
```

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

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

# Visualize the decomposed series
plot(GaringerOzone.daily.decomp)
```



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

# Visualize the decomposed series
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
library(Kendall)
# Run SMK test
monthly_ozone_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
monthly_ozone_trend
```

```
## tau = -0.1, 2-sided pvalue =0.16323
```

```
summary(monthly_ozone_trend)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

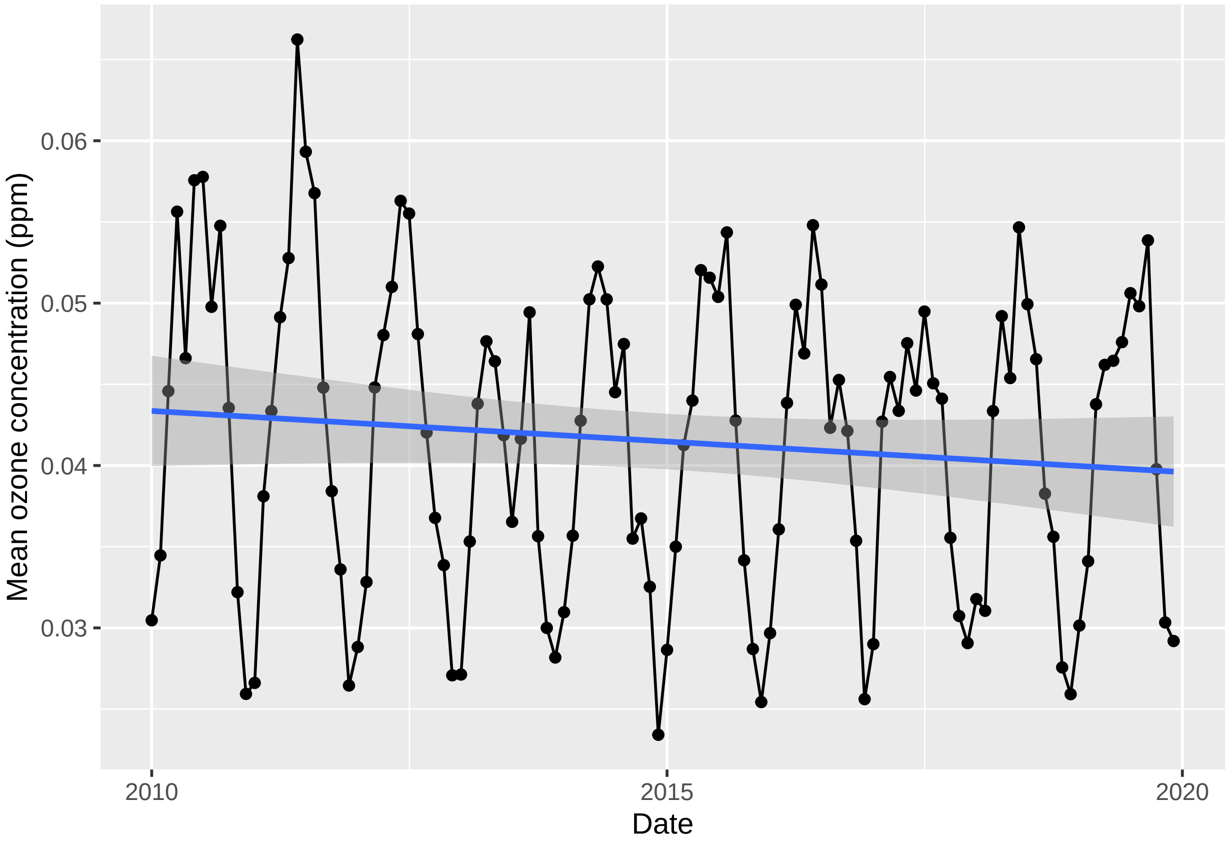
```
# p-value > 0.05, so fail to reject the null -> there is not a trend in the data
```

Answer: The seasonal Mann-Kendall is most appropriate because there is seasionality in our 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.monthly <-
  GaringerOzone.monthly %>%
  mutate(Date = lubridate::my(paste0(month,"-",year)))

#Visualization
monthly_ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean.ozone.conc)) +
  geom_point() +
  geom_line() +
  ylab("Mean ozone concentration (ppm)") +
  geom_smooth( method = lm )
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: While ozone concentrations may have changed over the 2010s at this station, there is not a significant trend in monthly ozone concentrations (p > 0.05).

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

# We can extract the components and turn them into data frames
GaringerOzone.monthly_components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,2:3])

#16

# adding an observed column and a date column to the data frame
GaringerOzone.monthly_components <- mutate(GaringerOzone.monthly_components,
        Ozone_conc = GaringerOzone.monthly$mean.ozone.conc,
        Date = GaringerOzone.monthly$Date)

# non-seasonal ozone
```

```
GaringerOzone.monthly.ns.ts <- ts(GaringerOzone.monthly_components$Ozone_conc, start = c(2010,1), freque

# Run MK test
monthly_ozone_trend1 <- Kendall::MannKendall(GaringerOzone.monthly.ns.ts)

# Inspect results
monthly_ozone_trend1
```

```
## tau = -0.105, 2-sided pvalue =0.088483
```

```
summary(monthly_ozone_trend1)
```

```
## Score =  -752 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.105, 2-sided pvalue =0.088483
```

```
# p-value > 0.05, so fail to reject the null -> there is not a trend in the data
```

Answer: While the p-value is closer to 0.05 than before, it is still greater than 0.05. Therefore, similar to the results obtained with the Seasonal Mann Kendall on the complete series, we fail to reject the null hypothesis and conclude that there is not a significant trend in monthly ozone concentrations (p > 0.05).