Assignment 8: Time Series Analysis

Julia Kagiliery

Spring 2025

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

```
getwd()
```

```
## [1] "/Users/juliakagiliery/Library/Mobile Documents/com~apple~CloudDocs/GitHub Links/EDAClas2025"
```

```
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 1.9.4     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)

theme_set(theme_minimal())
```

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

#here is where the CSVs are stored
csv_dir <- "Data/Raw/Ozone_TimeSeries"

# all the files I want = anything with a CSV extension in that folder
temp <- list.files(path = csv_dir,
                   pattern = "\\.csv$",
                   full.names = TRUE)

# Read CSV files into a list (couldn't figure out a better way to do this, this is annoying)
myfiles <- lapply(temp, read.csv)

# need to name my files, keep the name they had in folder. this sucks because it has the extesion still
names(myfiles) <- basename(temp)

# Check ut worked
print(names(myfiles))
```

```
##  [1] "EPAair_O3_GaringerNC2010_raw.csv" "EPAair_O3_GaringerNC2011_raw.csv"
##  [3] "EPAair_O3_GaringerNC2012_raw.csv" "EPAair_O3_GaringerNC2013_raw.csv"
##  [5] "EPAair_O3_GaringerNC2014_raw.csv" "EPAair_O3_GaringerNC2015_raw.csv"
##  [7] "EPAair_O3_GaringerNC2016_raw.csv" "EPAair_O3_GaringerNC2017_raw.csv"
##  [9] "EPAair_O3_GaringerNC2018_raw.csv" "EPAair_O3_GaringerNC2019_raw.csv"
```

```
#want data frames not list elements
list2env(myfiles, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

```r
#make the mega data set
GaringerOzone <- bind_rows(myfiles)

#pray this works
nrow(GaringerOzone)
```

```
## [1] 3589
```

```r
ncol(GaringerOzone)
```

```
## [1] 20
```

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

# 4
GaringerOzone <- GaringerOzone |>
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
# Generate 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"))
colnames(Days) <- "Date"

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

```
## Joining with `by = join_by(Date)`
```

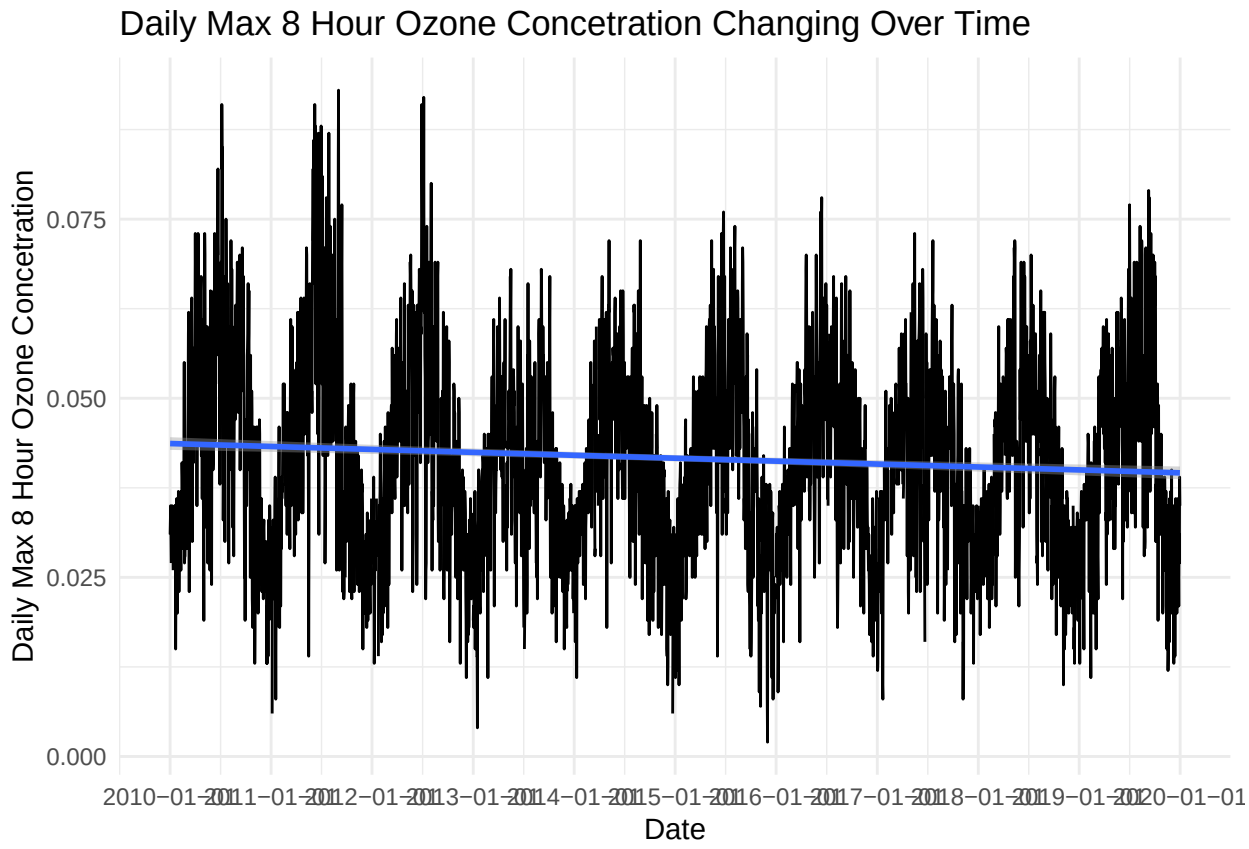
Visualize

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?

```r
#7
GaringerOzone |>
  ggplot(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(y = "Daily Max 8 Hour Ozone Concetration", title = "Daily Max 8 Hour Ozone Concetration Changing
  scale_x_date(date_breaks = "1 year")
```

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

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



Answer:There seems to be highly seasonal componets to ozone concentration over time (because it appears sinosodial) and slight decrease over time which is displayed by the downward slope of the linear trendline.

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?

```
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
```

```
#8
GaringerOzone <- GaringerOzone |>
  mutate(
    Daily.Max.8.hour.Ozone.Concentration = na.approx(Daily.Max.8.hour.Ozone.Concentration)
  )
```

```
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration) #so this does tell me that NA approx seems
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer:

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(
    Month = format(Date, "%m"),
    Day =  format(Date, "%d"), #not day is needed but
    Year = format(Date, "%Y")
  )

GaringerOzone.monthly <- GaringerOzone.monthly |>
  group_by(Month, Year) |>
  summarise(
    MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration)
  )
```

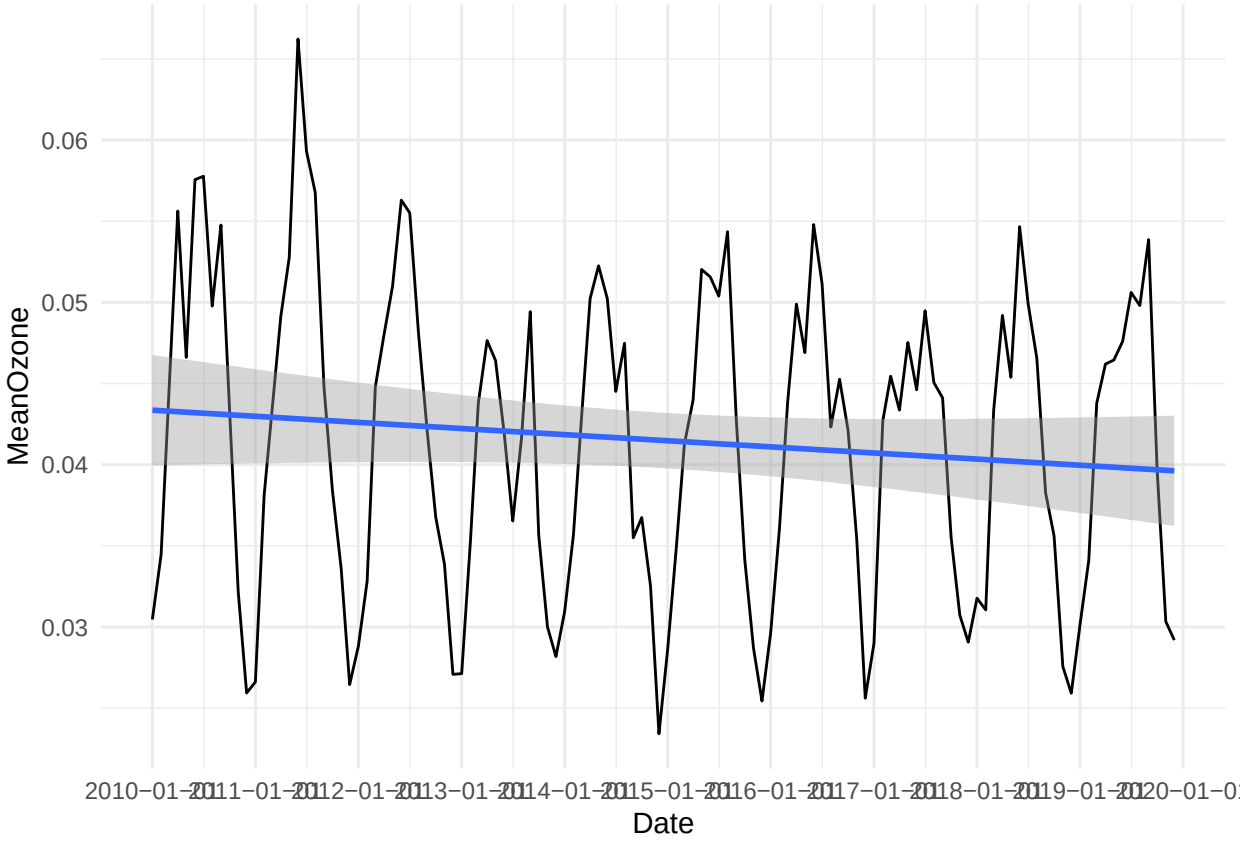
```
## `summarise()` has grouped output by 'Month'. You can override using the
## `.groups` argument.
```

```
GaringerOzone.monthly <- GaringerOzone.monthly |>
  mutate(
    Date = as.Date(paste(Year, Month, "01", sep = "-"))
  )

GaringerOzone.monthly$MeanOzone <- as.numeric(GaringerOzone.monthly$MeanOzone)
```

```
GaringerOzone.monthly |>
  ggplot(aes(x = Date, y = MeanOzone)) +
  geom_line() +
  geom_smooth(method = "lm") +
  scale_x_date(date_breaks = "1 year")
```

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




```
# I am just checking what the monthly stuff looks like, it looks good here but for some reason my times
```

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
min(GaringerOzone$Date)
```

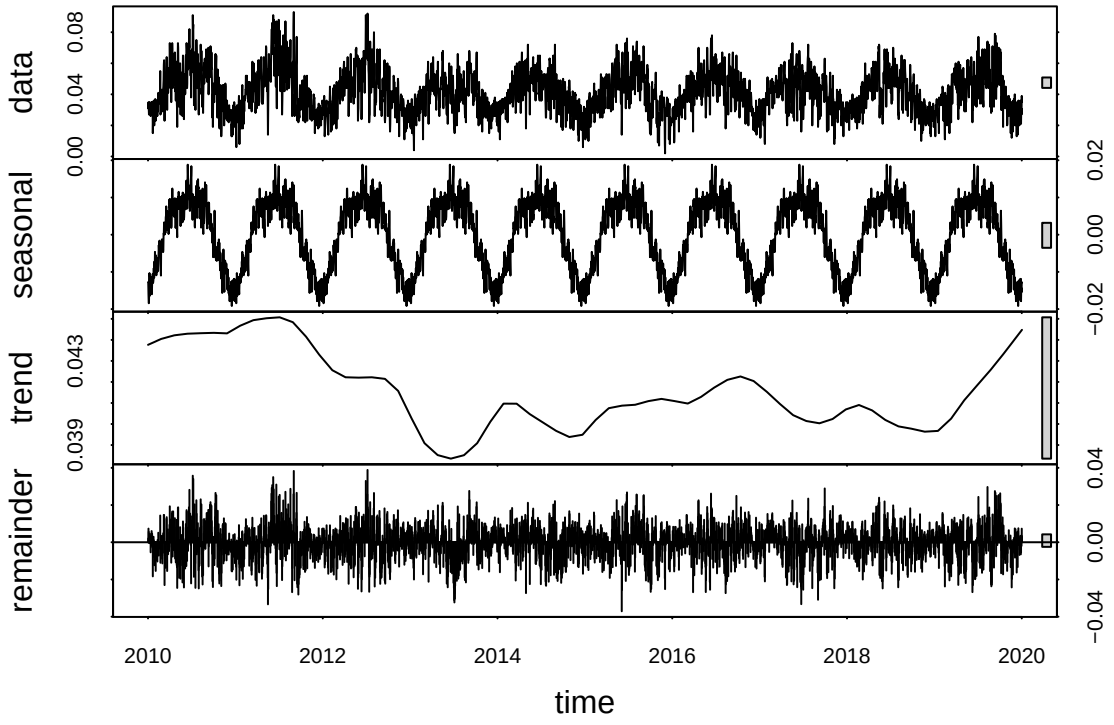
```
## [1] "2010-01-01"
```

```
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, start = c(2010, 1), fr

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone, start = c(2010, 1), frequency = 12) # s
```

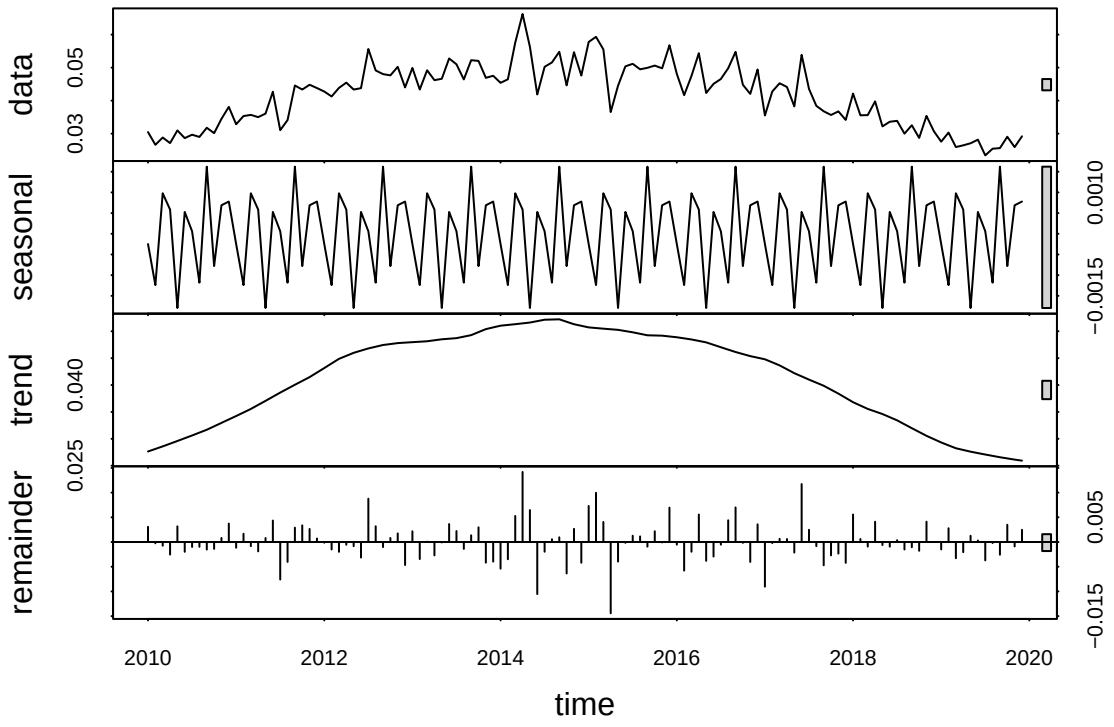
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11
GaringerOzone.daily.ts_decomp=stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.ts_decomp)
```




```r
GaringerOzone.monthly.ts_decomp = stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(GaringerOzone.monthly.ts_decomp) #okay I know this is incorrect and I have no idea why
```



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
monthly_trend<-trend::smk.test(GaringerOzone.monthly.ts)
summary(monthly_trend)
```

```
##
##  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  1.00000
## Season 2:   S = 0     5  125  0.111  0.358  0.72051
## Season 3:   S = 0    -3  125 -0.067 -0.179  0.85803
## Season 4:   S = 0     1  125  0.022  0.000  1.00000
## Season 5:   S = 0    -9  125 -0.200 -0.716  0.47427
## Season 6:   S = 0     1  125  0.022  0.000  1.00000
## Season 7:   S = 0   -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0    -3  125 -0.067 -0.179  0.85803
```

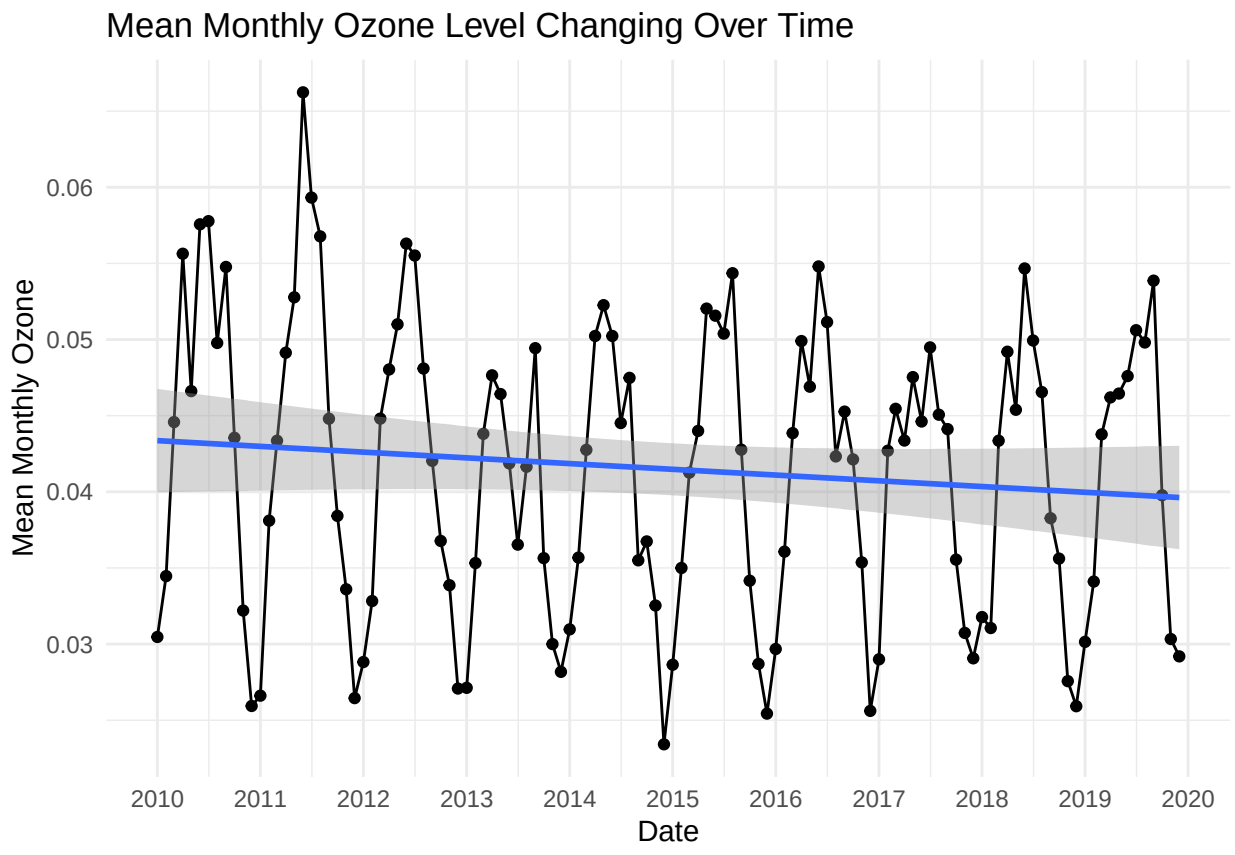
```
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -11  125 -0.244 -0.894  0.37109
## Season 11:   S = 0 -15  125 -0.333 -1.252  0.21050
## Season 12:   S = 0  -5  125 -0.111 -0.358  0.72051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The SMK test is the best because it accounts for seasonality and is non-parametruc. Other tests don't work as well 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
GaringerOzone.monthly |>
  ggplot(aes(x = Date, y = MeanOzone)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(
    y= "Mean Monthly Ozone",
    title = "Mean Monthly Ozone Level Changing Over Time"
  )
```

```
## `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: There is not sufficent evidence to reject the null hypothesis the the ozone levels have changed over the 2010s (The lowest p-value is 0.21050). In the context of the figure, we see

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
SeasonalComponent <-GaringerOzone.monthly.ts_decomp$time.series[, "seasonal"]
ozone_monthly_noseason <- GaringerOzone.monthly.ts- SeasonalComponent

#16
monthly_trend_noseas <- trend::mk.test(ozone_monthly_noseason)
summary(monthly_trend_noseas)
```

```
##             Length Class  Mode
## data.name   1      -none- character
## p.value     1      -none- numeric
## statistic   1      -none- numeric
## null.value  1      -none- numeric
## parameter   1      -none- numeric
## estimates   3      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## pvalg       1      -none- numeric
```

Answer: So my time series is not working, but I would immagine the point of these two questions is to show us that if we remove seasonality, there is now evidence to reject the null hypothesis and say there is a decreasing trend. This appears to confirm what can be visually inspected in the graphs via the trendline.