Assignment 8: Time Series Analysis

Brock Keller

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

```
library(here)
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(dplyr)

getwd()
```

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

```
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
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.

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")
class(GaringerOzone$Date) #checking that I got the format right
```

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

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

# 5
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))

colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
```

Visualize

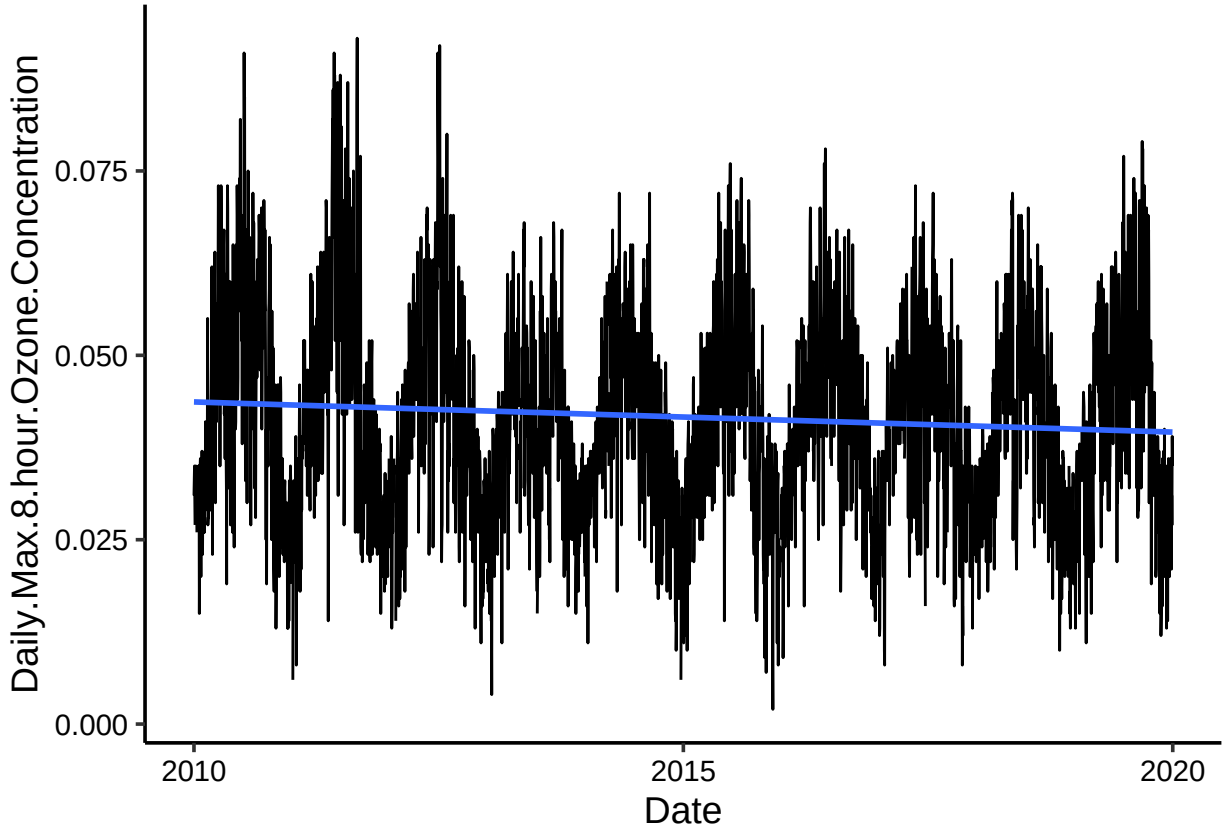
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
Ozone_over_time <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method = "lm", se = FALSE)

print(Ozone_over_time)
```

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

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



Answer: The plot shows strongly seasonal data with an overall slowly decreasing trend in mean ozone concentration.

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$Daily.Max.8.hour.Ozone.Concentration <- na.approx(
  GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, x = GaringerOzone$Date)
```

Answer: A piecewise constant would just carry forward the last known value. Since our data gaps arent super huge (from a quick glance), this probably wouldn't destroy the integrity of the dataset, but it is very simplistic and a linear interpolation will work better for small gaps in relatively regular value fluctuations. Similarly, the data gaps are so small that there is not really need for a spline interpolation fitting a whole curve to it and making things overly complicated.

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(Date),
    month = month(Date),
    DateNew = as.Date(paste(year, month, 1, sep = "-"))
  ) %>%
  group_by(year, month) %>%
  summarize(
    mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE)
  ) %>%
  mutate(
    DateNew = as.Date(paste(year, month, 1, sep = "-"))
  )
```

```
## `summarise()` has grouped output by 'year'. 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.

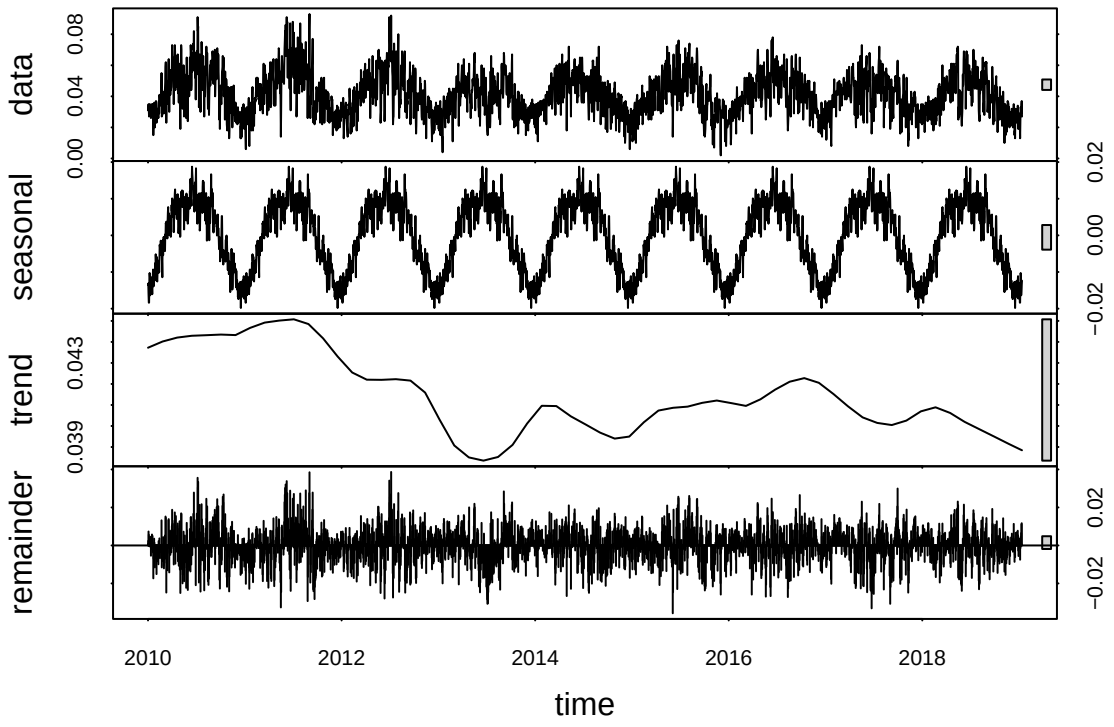
```
#10
GaringerOzone.daily.ts <- ts(
  GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
  start = c(2010, 1),
  end = c(2019, 12),
  frequency = 365
)

GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$mean_ozone,
  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.

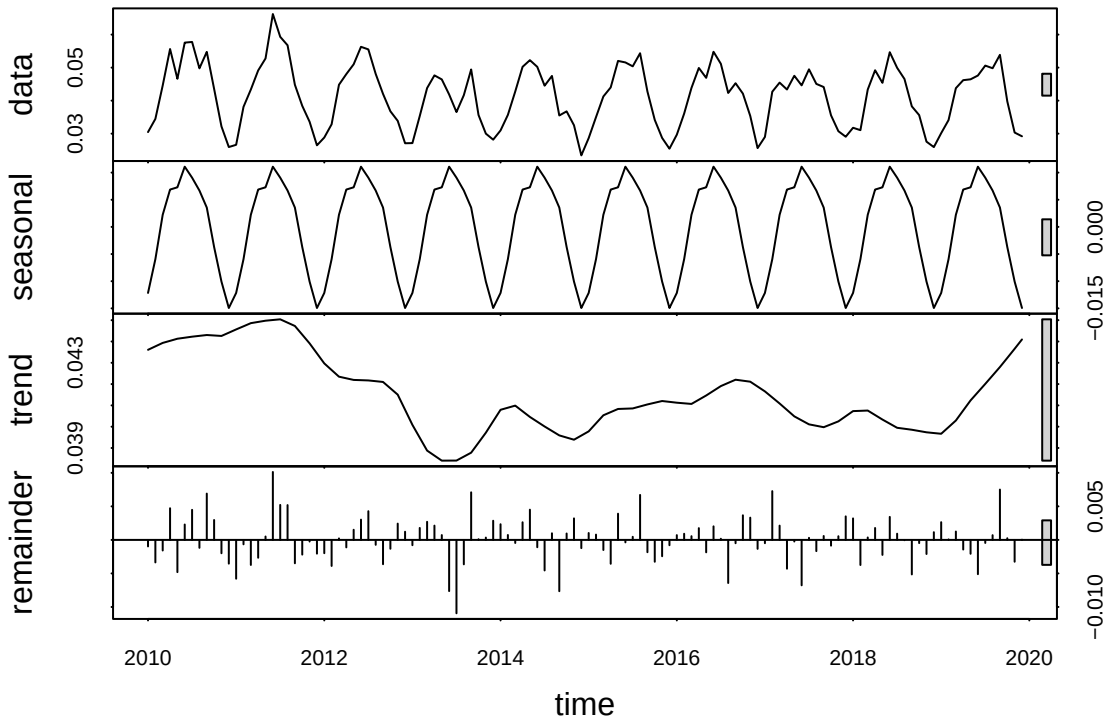
```
#11
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.decomposed)
```




```
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(GaringerOzone.monthly.decomposed)
```



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.Ozone.trend <- trend::smk.test(GaringerOzone.monthly.ts)

Monthly.Ozone.trend
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

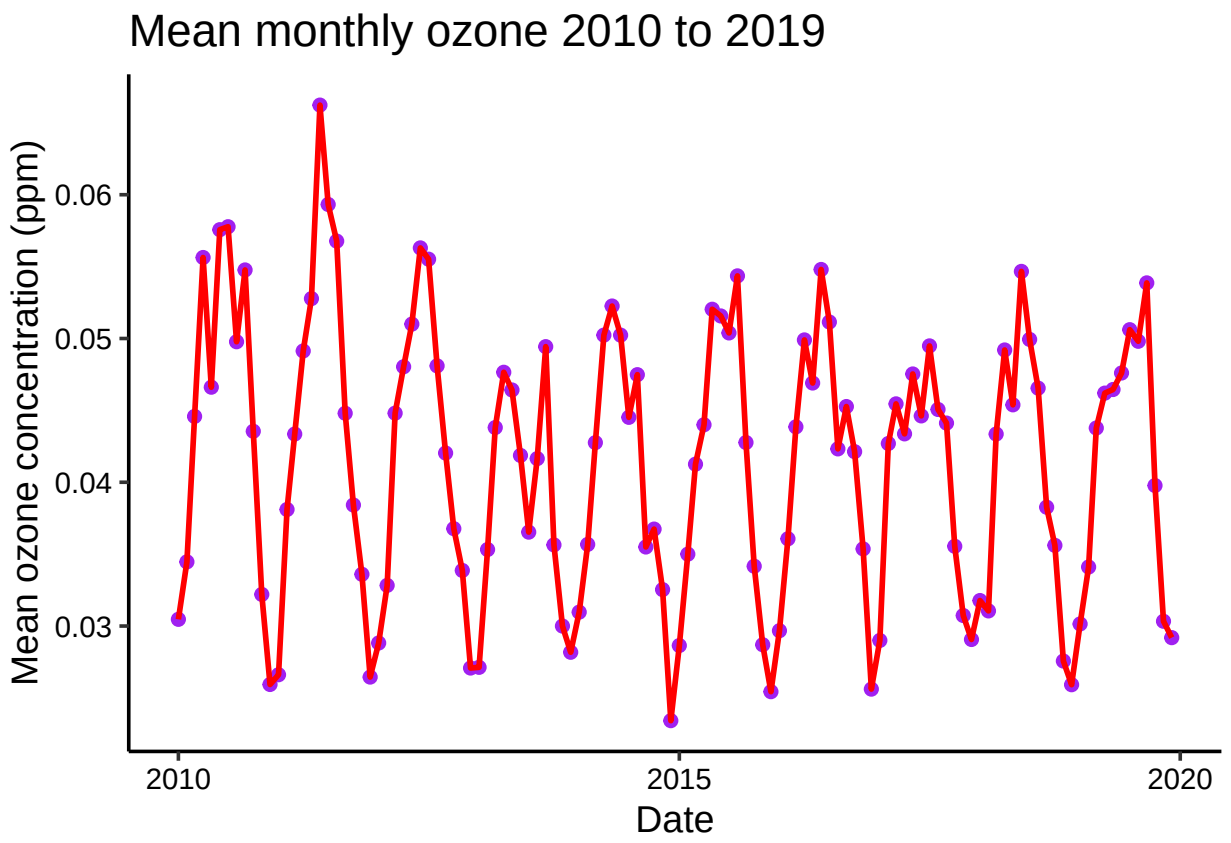
Answer: The seasonal Mann-Kendall test is most appropriate because our ozone data clearly shows a strong seasonal trend. A normal Mann-Kendall test would not recognize the repeat seasonality which would throw the interpretation off. The seasonal test does multiple analyses to look for trends between each seasonal cycle instead of the whole thing as a monolith.

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
monthly_ozone_over_time <- ggplot(GaringerOzone.monthly, aes(x = DateNew, y = mean_ozone)) +
  geom_point(color = "purple", size = 2)+
  geom_line(color = "red", size = 1)+
  labs(
    title = "Mean monthly ozone 2010 to 2019",
    x = "Date",
    y = "Mean ozone concentration (ppm)"
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
print(monthly_ozone_over_time)
```



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 seasonal Mann-Kendall trend test showed that there is indeed a change in ozone concentrations over the 2010s at this site. The graph allows you to see it somewhat, but the

test results help make the visualization more clear. From 2010 to 2019, mean monthly ozone concentrations have been slightly decreasing, and the p value is significant enough that we can reject the null hypothesis, barely (z = -1.963 showing negative trend, p value = 0.04965 showing statistical significance, S = -77 meaning trend is decreasing, varS is 1499).

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
GaringerOzone.monthly.components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series) %>%
  select(trend, remainder)


#16
GaringerOzone.monthly.deseazonized <- trend::mk.test(GaringerOzone.monthly.components$trend)

GaringerOzone.monthly.deseazonized
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.components$trend
## z = -4.3573, n = 120, p-value = 1.317e-05
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.922000e+03  1.943667e+05 -2.691877e-01
```

Answer: The non-seasonal monthly series also shows a decrease and statistical significance, but much sharper ones than the seasonal data. z = -4.3573, p value = 1.317eˆ-05.