Assignment 8: Time Series Analysis

Ellie Harrigan

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(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(readr)
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```
#install.packages("trend")
#install.packages("zoo")
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
##
## Attaching package: 'here'
##
## The following object is masked from 'package:plyr':
##
##     here
```

```
getwd()
```

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

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.

```
#2

GaringerFiles <- list.files(path = "./Data/Raw/Ozone_TimeSeries/",
                            pattern = "*.csv",
                            full.names = TRUE)
GaringerFiles
```

```
##  [1] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2010_raw.csv"
##  [2] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2011_raw.csv"
##  [3] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2012_raw.csv"
##  [4] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2013_raw.csv"
##  [5] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2014_raw.csv"
##  [6] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2015_raw.csv"
##  [7] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2016_raw.csv"
##  [8] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2017_raw.csv"
##  [9] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2018_raw.csv"
## [10] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2019_raw.csv"
```

```
GaringerOzone <- GaringerFiles %>%
  plyr::ldply(read.csv)
```

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")

GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4
GaringerOzoneWrangle <- 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, GaringerOzoneWrangle, by = c("Date"))
```

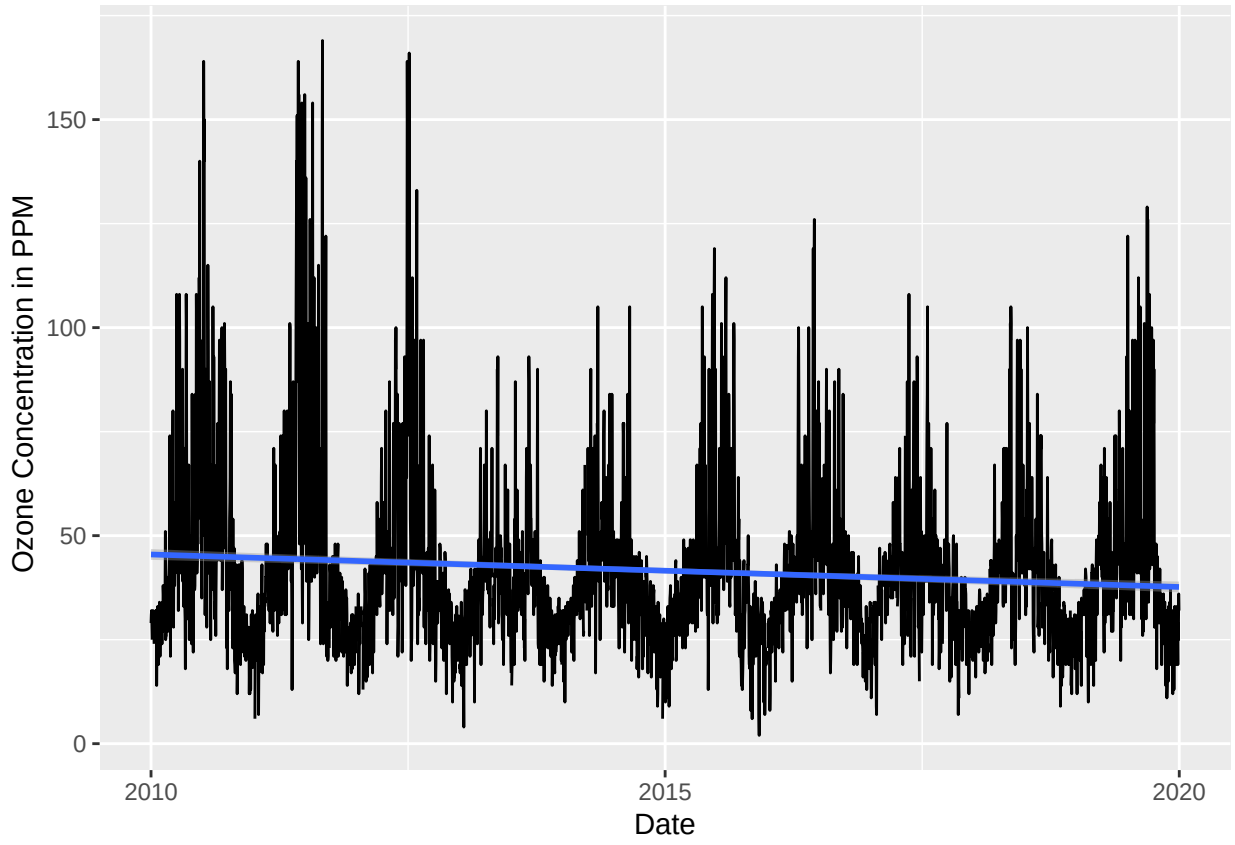
Visualize

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_AQI_VALUE)) +
  geom_line() +
  labs(y = " Ozone Concentration in PPM") +
geom_smooth(method = "lm")
```

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

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



Answer: Slight decrease in ozone concentrations as time increases.

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

summary(GaringerOzone_Clean$Ozone.clean)
```

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

Answer: We used the linear interpolation method because it connects the dots by assuming the missing data falls between the previous and next measurement. This method doesn't change the min, max, or mean values of the dataset, therefore it doesn't change the main properties of the distribution. The piecewise constant because any missing data is assumed to be equal to the nearest measurement, thus assuming the value does not change between measurements which is unlikely in the case of ozone and AQI data. The spline interpolation it uses a quadratic function rather than linear line to inerpolate and would be better used for non-linear 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)

#For some reason when I run this code it gives me 1 observation and 1 variable for the GaringerOzone.monthly, I worked on it for about 2 hours and couldn't figure out why it wasn't working. I have a feeling it has something to do with my date class and the date column not being recognized properly when I mutate, then group_by, and when I summarise the meanOzone it kept giving me 1 observation and 1 variable. I tried to answer questions 10-15 to the best of my ability without using this monthly data frame, apologies since I know they are wrong and I couldn't complete #15 and #16. I'll stop by someone's office hours next week to sort out why I couldn't get this code chunk to work.

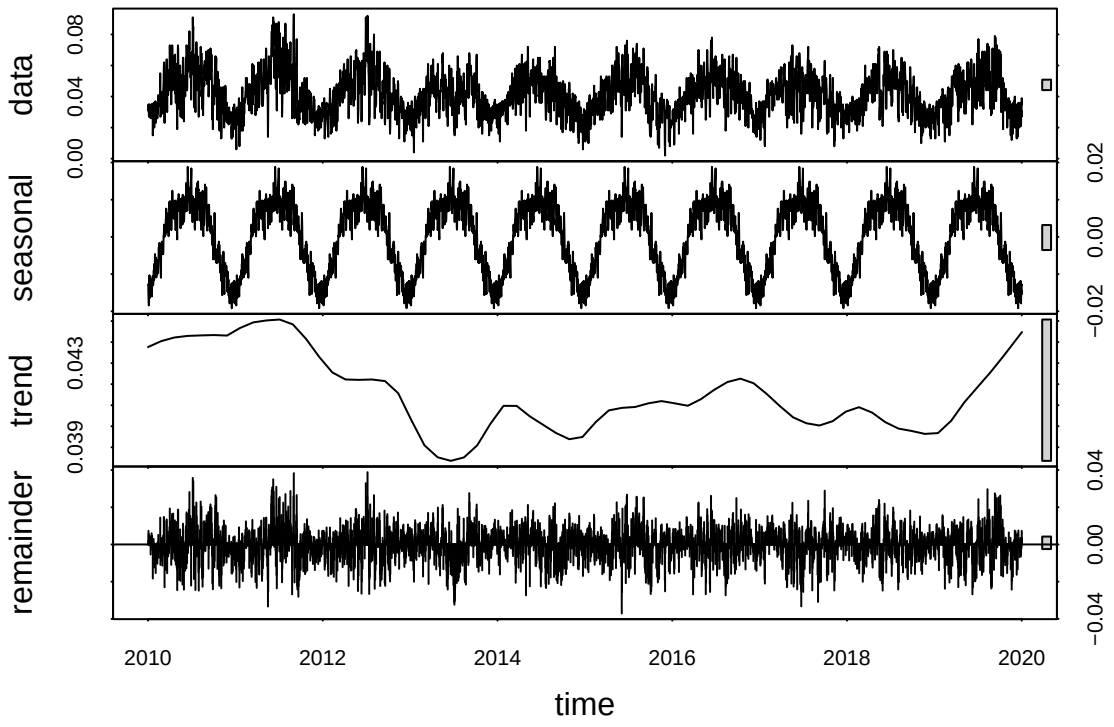
```
#9
#GaringerOzone.monthly <- GaringerOzone %>%
 # mutate(year = year(Date),
  #       month = month(Date)) %>%
  #group_by(month) %>%
  #summarise(meanozone = mean(Daily.Max.8.hour.Ozone.Concentration))
 # mutate( Date = my(paste0(month,"-",year)))
```

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_Clean$Ozone.clean, start = c(2010,1),
                             frequency = 365)
#GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanOzone, start =
#c(2010,1),
#frequency = "periodic")
```

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)
```

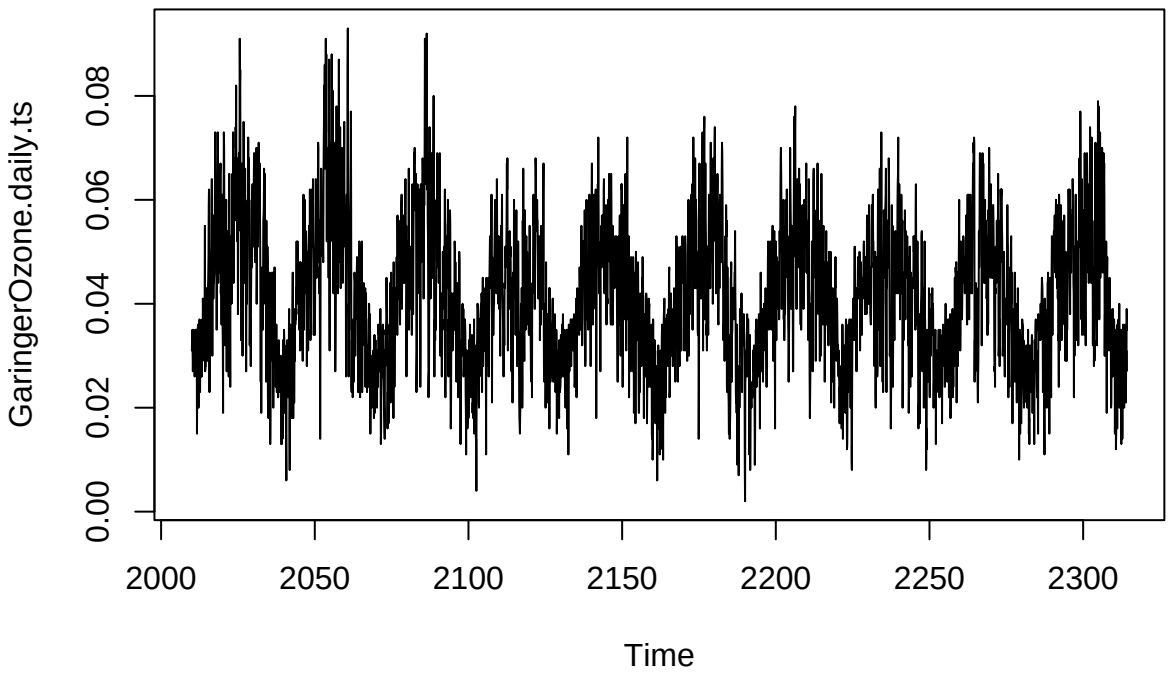



```
#GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts,
                                         #s.window > "1")
#plot(GaringerOzone.monthly.decomposed)
```

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
f_month <- month(first(GaringerOzone_Clean$Date))
f_year <- year(first(GaringerOzone_Clean$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone_Clean$Ozone.clean,
                             start = c(f_year, f_month),
                             frequency = 12)

garinger_ozone_decompose <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.ts)
```




```
garinger_ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.daily.ts)

summary(garinger_ozone_trend1)
```

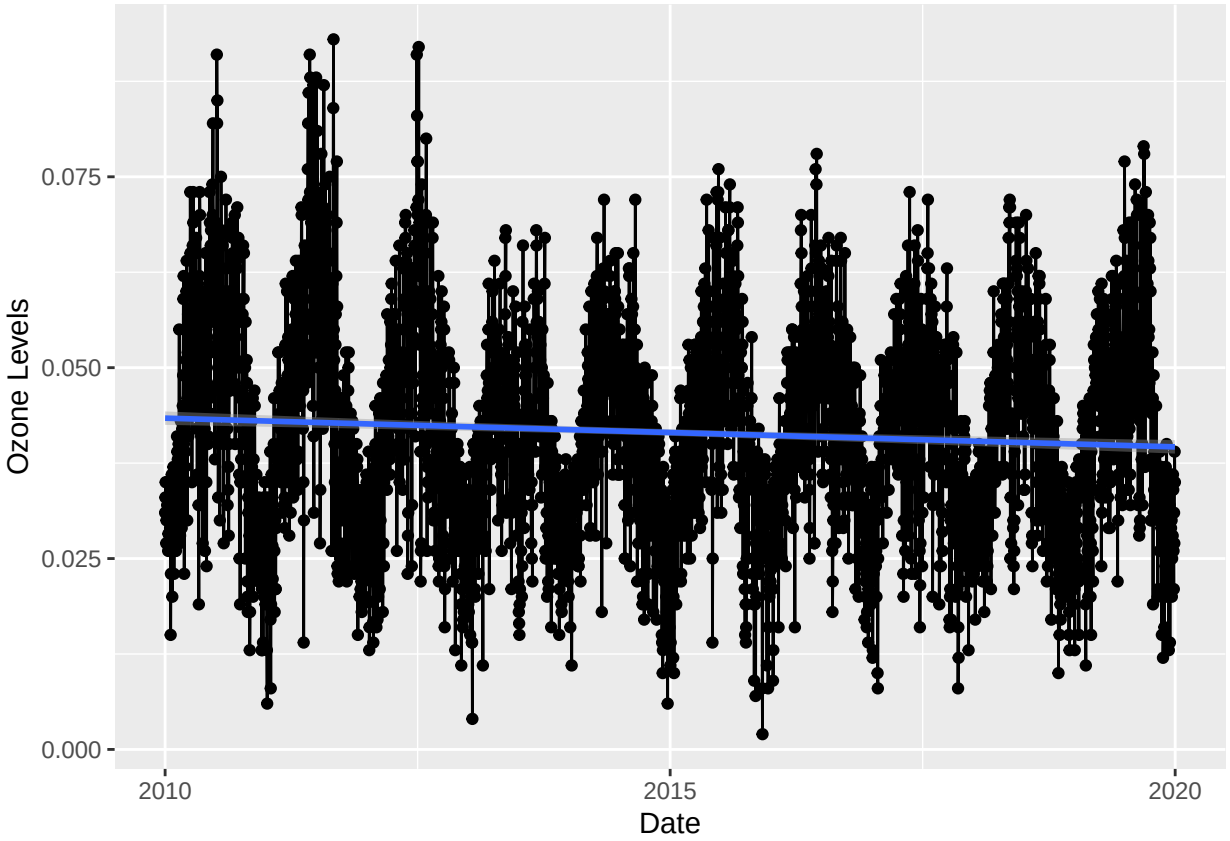
```
## Score =  -22362 , Var(Score) = 37738106
## denominator =  548228
## tau = -0.0408, 2-sided pvalue =0.00027247
```

Answer: The Mann-Kendall is most appropriate because it can account for the inherent seasonality changes (temperature and precipitation) that comes with monthly ozone levels.

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
garinger_ozone_plot <- ggplot(GaringerOzone_Clean, aes(x = Date,
                                                       y = Ozone.clean )) +
  geom_point() +
  geom_line() +
geom_smooth(method = "lm") +
labs(x = "Date",
       y = "Ozone Levels")
print(garinger_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: The plot shows the monthly ozone levels (represented by the clean ozone concentration values) over time. The data points are connected with a line to visualize the changes in ozone concentrations, with a linear regression line (in blue) set to assess the overall trend. From the graph, it appears there may be a decreasing trend in ozone levels over the study period. A Seasonal Mann-Kendall test was performed To assess whether there is a statistically significant trend in the ozone levels over time. The test statistic (tau = -0.0408) indicates a slight negative trend in ozone concentrations over the study period. This negative value suggests that ozone levels are slightly decreasing on average over time. The p-value (0.00027247) is statistically significant, illustrating there is evidence to reject the null hypothesis of no trend, showing that the observed trend is unlikely to be due to random chance.

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



#16
```

Answer: