SamanthaSedar_A08_TimeSeries

Samantha Sedar

Fall 2023

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
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2023
```

```
library(cowplot)
library(agricolae)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.3     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x lubridate::stamp() masks cowplot::stamp()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

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

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

```
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Set theme
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.

```
#2

O10 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFa
```

```
O11 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFa
O12 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFa
O13 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFa
O14 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFa
O15 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFa
O16 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFa
O17 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFa
O18 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFa
O19 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFa


GaringerOzone <- rbind(O10, O11, O12, O13, O14, O15, O16, O17, O18, O19)
```

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.

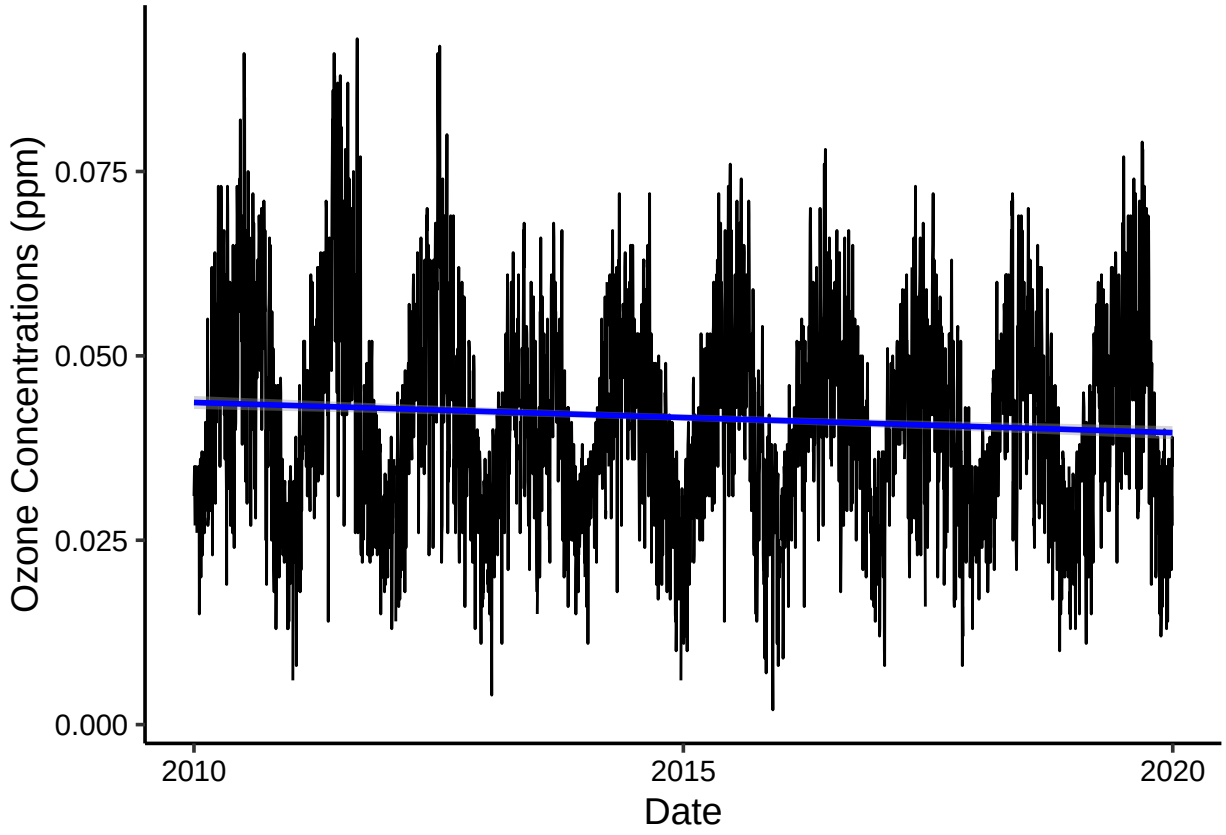
```
# 3- setting as date using tidyverse

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

# 4- Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentrati

GaringerOzone_Processed <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_

# 5
# data frame with a sequence of dates
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
Days <- as.data.frame(seq(start_date, end_date, by = 1))


# Rename the column in Days to "Date"
colnames(Days) <- "Date"

# 6

# Left join Days with the combined data frame
GaringerOzone <- left_join(Days, GaringerOzone_Processed, 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

ggplot(GaringerOzone, aes(x= Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm, color="blue")+
  labs(x = "Date", y = "Ozone Concentrations (ppm)")
```

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

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



Answer: The concentrations appear to have a seasonal trend, with the smoothed linear line indicating a gradual decrease between 2010 and 2020.

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

#na approx
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  zoo::na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

Answer: In piecewise constants, missing data are assumed to be equal to the measurement made nearest to that date, which wouldnt apply. Similarly this isnt a quadratic function, therefor spline would also not apply.

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 [collaborated on this one to get the grou_by /ungroup piece]


GaringerOzone.monthly <-
  GaringerOzone %>%
   mutate(Month=month(Date), Year=year(Date))%>%
  group_by(Year, Month) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  ungroup()%>%
mutate(Date = make_date(Year, Month))
```

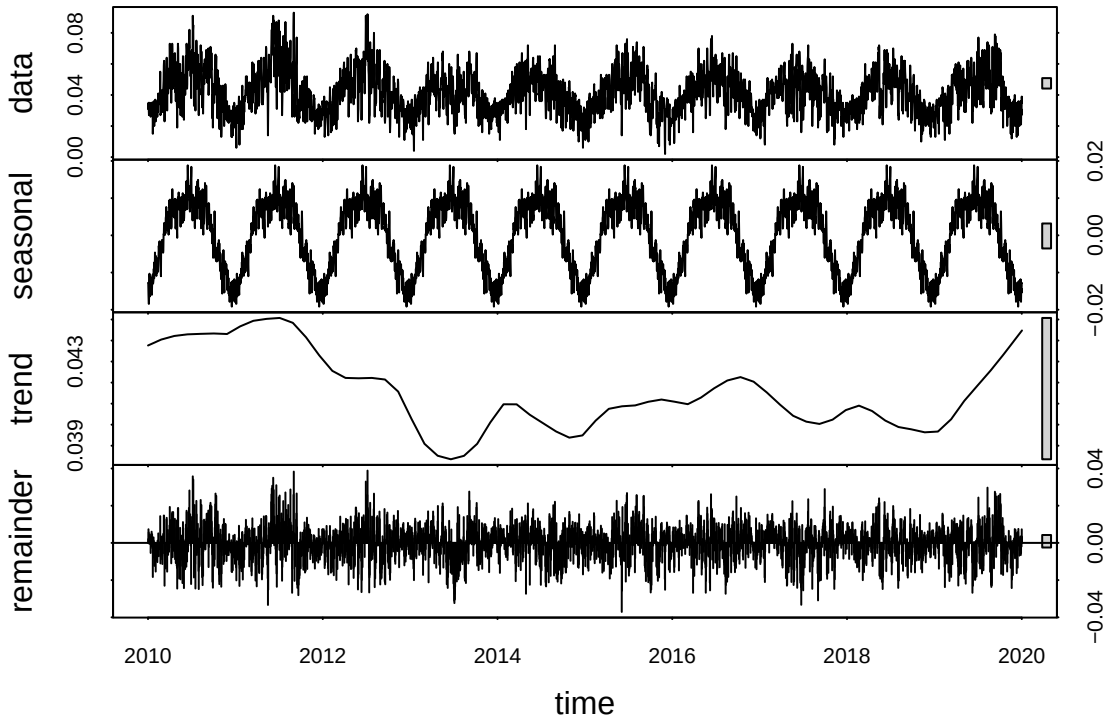
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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

#Daily observations, using either daily or monthly determined by frequency
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 1), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 1), frequency = 12)
```

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

```r
#11

#decompose daily
daily_stl <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(daily_stl)
```




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

12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```r
#12

#smk test
trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

#display
trend1
```

```
## tau = -0.0408, 2-sided pvalue =0.00027247
```

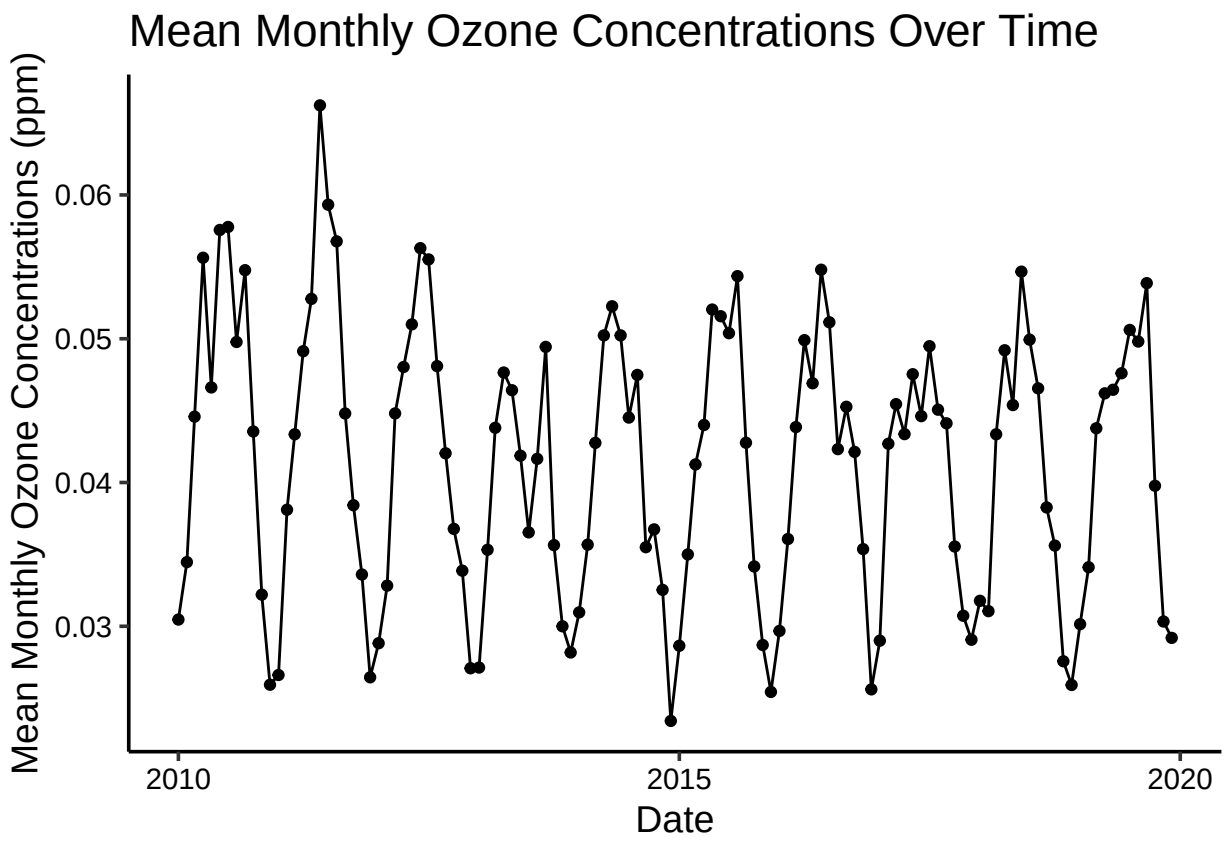
```
summary(trend1)
```

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

Answer: When we first ran our plot, we determined that the observations displayed a seasonal trend and the Seasonal Mann-Kendall test is the only test we have to examine seasonal trends.

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
# ggplot
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Monthly Ozone Concentrations (ppm)") +
  ggtitle("Mean Monthly Ozone Concentrations Over Time")
```



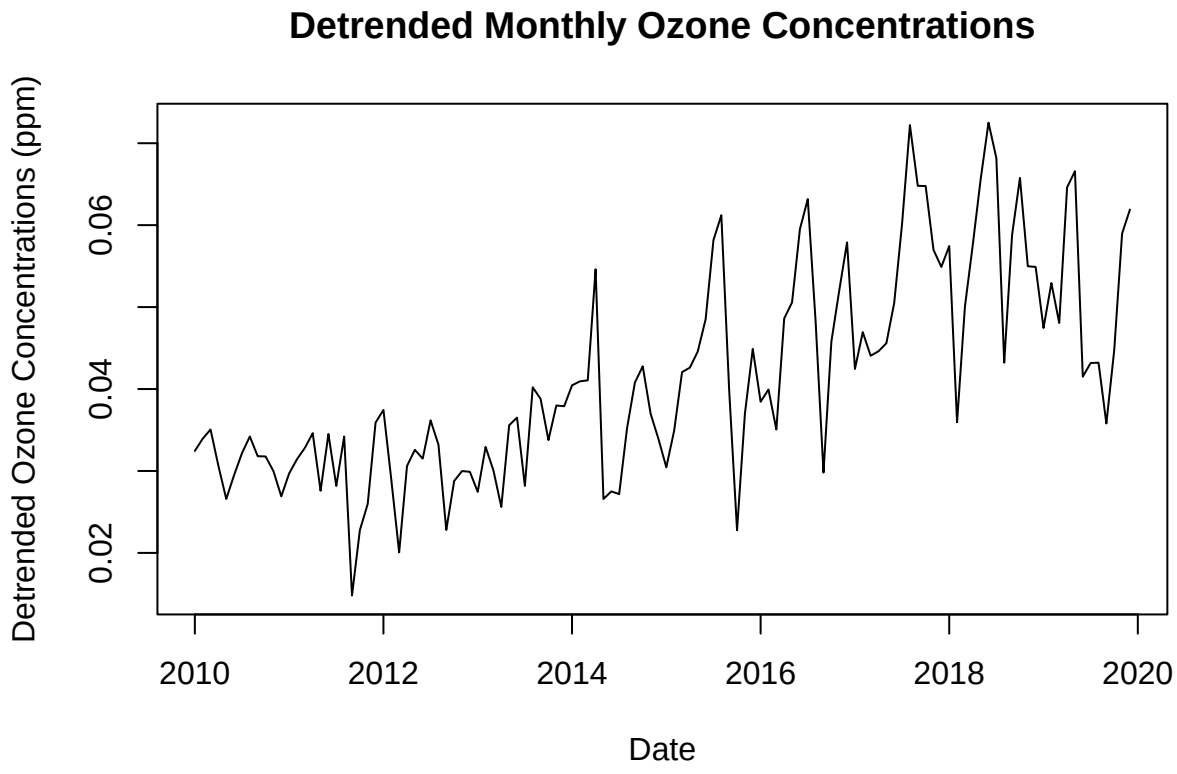
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

The negative tau value of -0.0408 suggests that ozone concentrations have exhibited a decreasing trend over the 2010s. This trend is statistically significant as indicated by the p-value (p = 0.00027247), which is less than the significance level of 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.

```r
#15

#decompose
decomp <- GaringerOzone.monthly.ts - monthly_stl$time.series[, "seasonal"]

GaringerOzone_monthly_detrended_ts <- ts(decomp,
                                         start = c(2010, 1),
                                         end = c(2019, 12),
                                         frequency = 12)

#plot
plot(GaringerOzone_monthly_detrended_ts, main = "Detrended Monthly Ozone Concentrations",
     xlab = "Date", ylab = "Detrended Ozone Concentrations (ppm)")
```



```r
#16
# mann-kendall on original
non_seasonal_trend <- Kendall::MannKendall(GaringerOzone_monthly_detrended_ts)
summary(non_seasonal_trend)
```

```
## Score =  3946 , Var(Score) = 194356.7
## denominator =  7134.999
## tau = 0.553, 2-sided pvalue =< 2.22e-16
```

Answer: Both tests indicate a statistically significant decreasing trend in the mean of monthly ozone concentrations.