EmmaBeyer_A08_TimeSeries.Rmd

Emma Beyer

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

```
# loading packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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(dplyr)
library(lubridate)
library(trend)
library(zoo)
```

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

```
library(here)
```

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

```
# check WD
here()
```

```
## [1] "/home/guest/EDA/EDE_Fall2023"
```

```
# create my theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "darkgreen"),
        title = element_text(color='darkblue'),
        legend.position = "right")
# set theme
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
# load each dataset
garinger_2010 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2011 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2012 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2013 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2014 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2015 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2016 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
           stringsAsFactors = TRUE)
```

```
garinger_2017 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2018 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
           stringsAsFactors = TRUE)
garinger_2019 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
           stringsAsFactors = TRUE)

# combine all the datasets
GaringerOzone <- rbind(garinger_2010, garinger_2011, garinger_2012,
                       garinger_2013, garinger_2014, garinger_2015,
                       garinger_2016, garinger_2017, garinger_2018,
                       garinger_2019)
```

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 date class
  GaringerOzone$Date <- mdy(GaringerOzone$Date)

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

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

# 6
# combine new date column with dataset to fill missing values with NAs
GaringerOzone2 <- left_join(Days, GaringerOzone_processed)
```

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

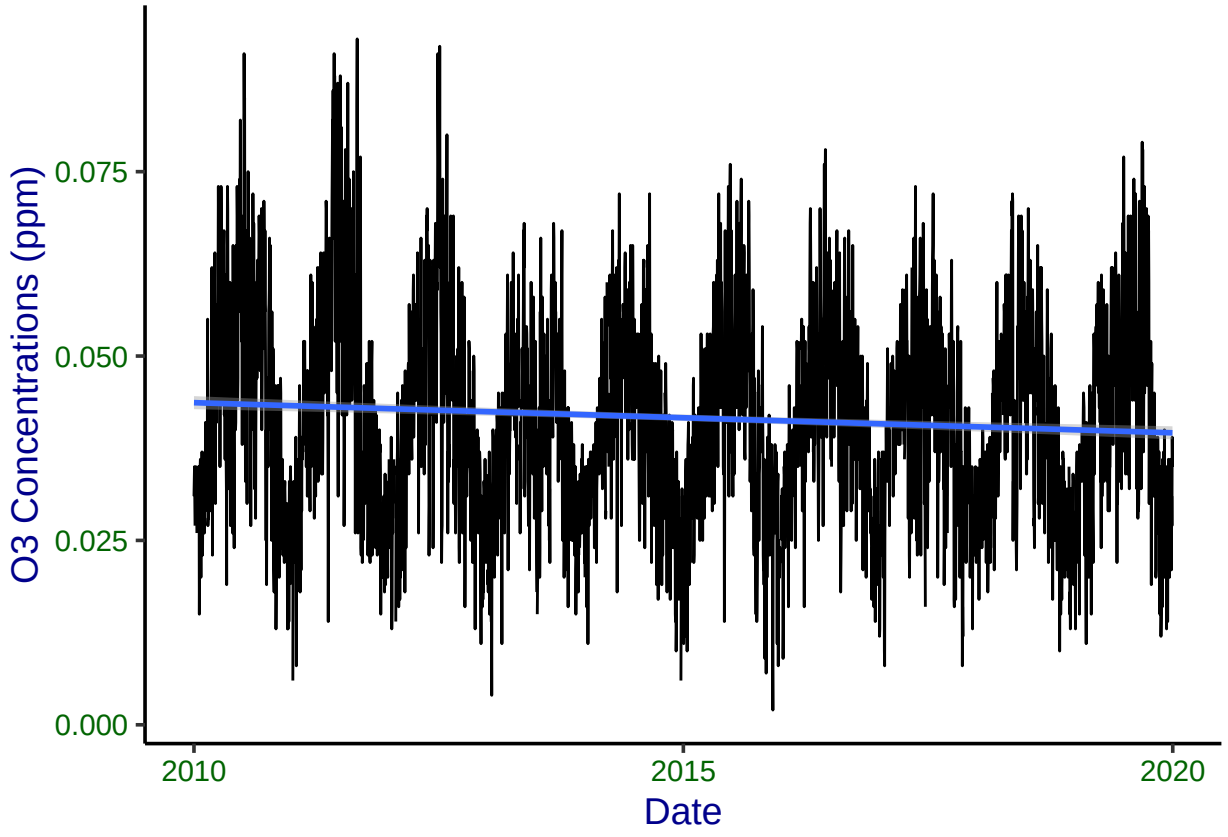
Visualize

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?

```
#7
# plot ozone concentrations
ozone_plot <-
ggplot(GaringerOzone2,
       aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(x = "Date", y = "O3 Concentrations (ppm)")
print(ozone_plot)
```

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

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



Answer: The concentration mostly remains the same, maybe decreases a little.

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
# linear interpolation of ozone concentrations
GaringerOzone_fill <-
  GaringerOzone2 %>%
  mutate( Daily.Max.8.hour.Ozone.Concentration =
          zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
```

Answer: Linear will not effect the trend of the missing values, and create linear lines between the missing values. The other two use quadratic equations, which will change how our data is represented and skew the results.

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
# create year and month columns
GaringerOzone_fill$year <- year(GaringerOzone_fill$Date)
GaringerOzone_fill$month <- month(GaringerOzone_fill$Date)

GaringerOzone.monthly <-
  GaringerOzone_fill %>%
  # grouping by year then month
  group_by(year, month) %>%
  # create mean ozone column
  summarise(mean_ozone=mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  # sort by first day of month
  mutate(Day="01") %>%
  # add date column back into data frame
  mutate(Date=as.Date(paste(year, month, Day, sep="-"), "%Y-%m-%d"))
```

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

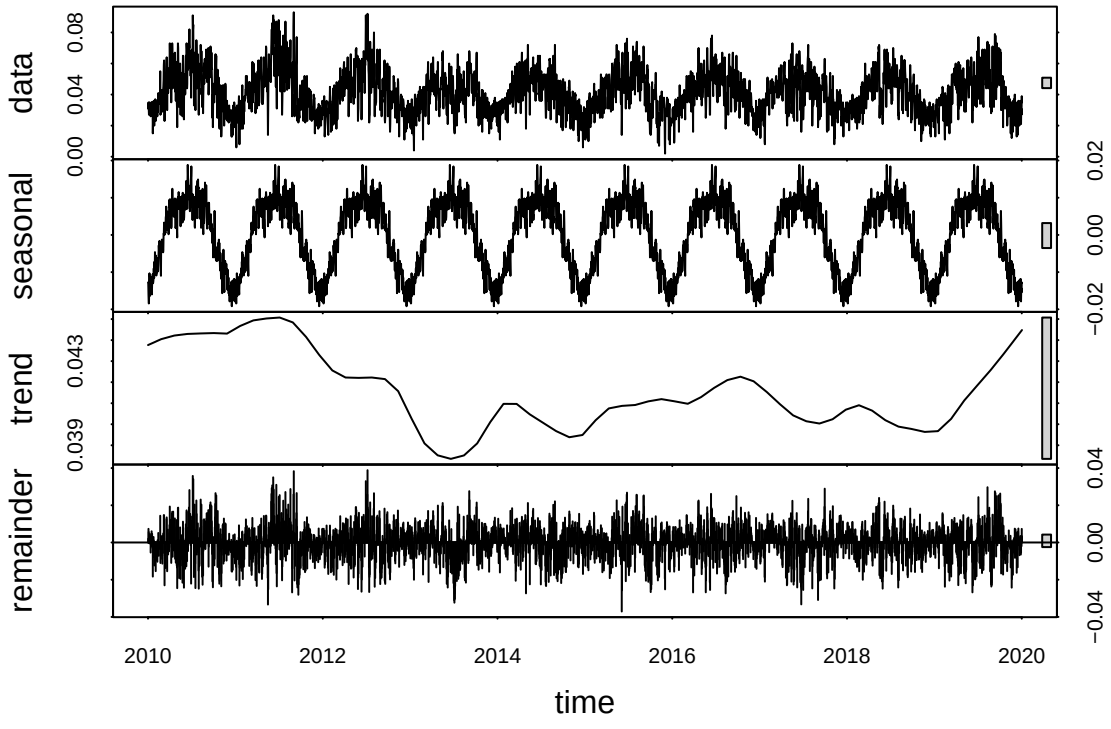
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10
# time series of daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone_fill$Daily.Max.8.hour.Ozone.Concentration,
                      start = c(2010,1),
```

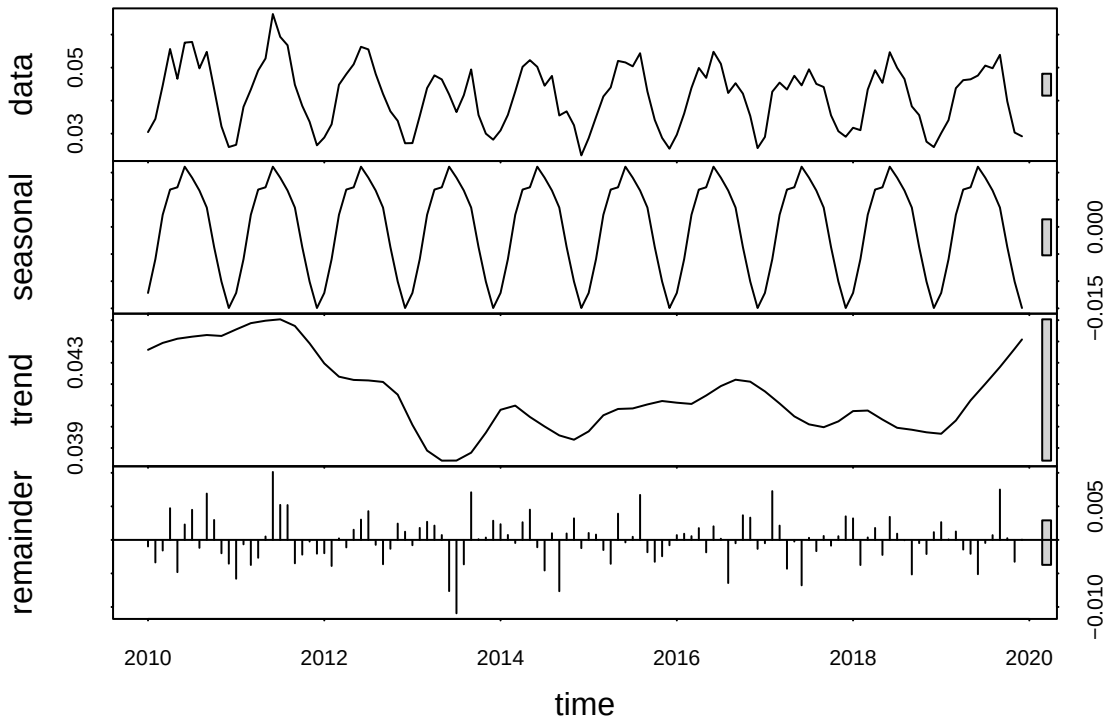
```
                        frequency = 365)

# time series of monthly observations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                        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
# decompose of daily observations
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts,
                                      s.window = "periodic")
plot(GaringerOzone.daily.decomposed)
```




```
# decompose of monthly observations
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
# run seasonal MannKendall
GaringerOzone_monthly_trend <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
# summarize results
summary(GaringerOzone_monthly_trend)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

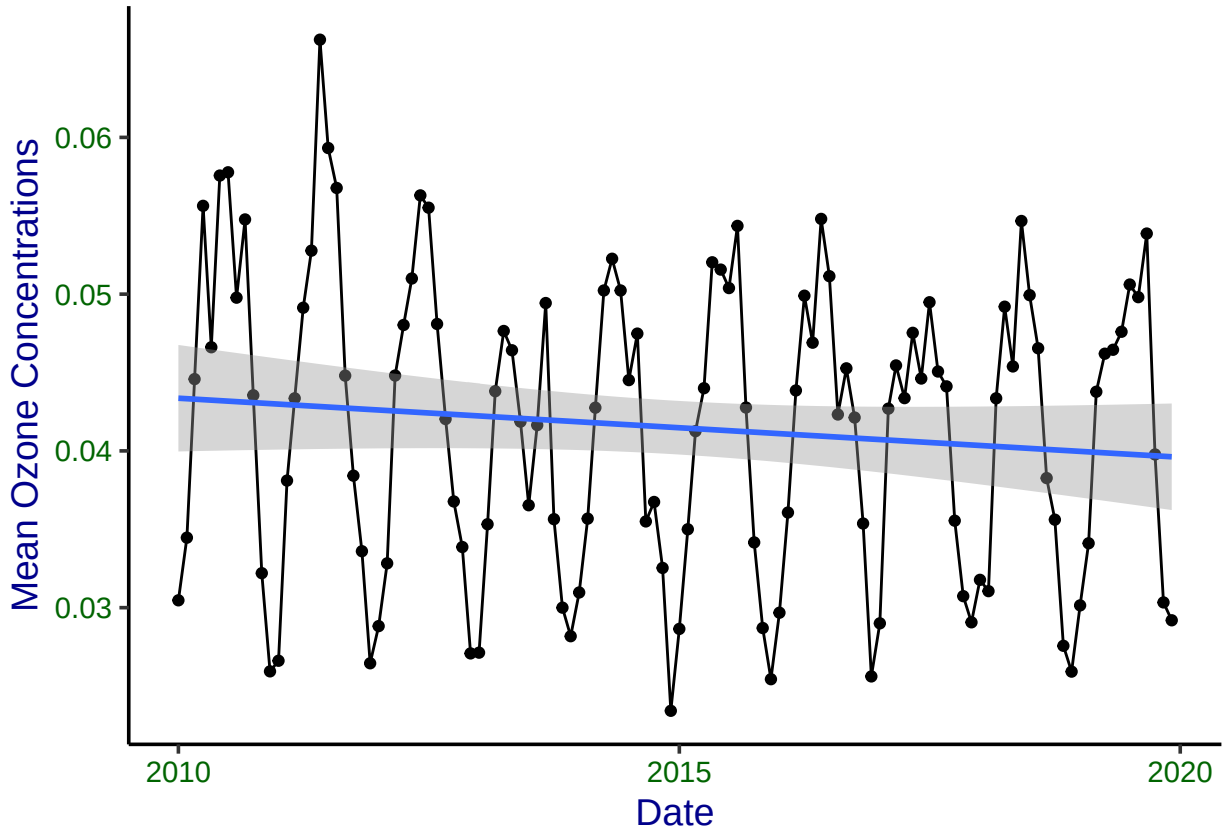
Answer: Based on our results from the decomposed plots, this data is seasonal and non-parametric, so it makes sense to use the seasonal Mann-Kendall.

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
# plot monthly data frame
GaringerOzone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
```

```
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Ozone Concentrations") +
  geom_smooth( method = lm )
print(GaringerOzone.monthly.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: Ozone concentrations have decreased in the past 10 years. The p-value shows that there was a significant change (pvalue = 0.046724, tau = -0.143).

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
# wrangle data to exclude seasonal
GaringerOzone.monthly.components <-
```

```
  # make a time series
  as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

GaringerOzone.monthly.components <-
  # create observed, date, and nonseasonal columns
  mutate(GaringerOzone.monthly.components,
        Observed = GaringerOzone.monthly$mean_ozone,
        Date = GaringerOzone.monthly$Date,
        Nonseasonal = Observed - seasonal)

#16
# create time series of non seasonal
GaringerOzone.monthly.components.ts <- ts(GaringerOzone.monthly.components$Nonseasonal,
                      start = c(2010,1),
                      end = c(2019,12),
                      frequency = 12)
# run MannKendall
GaringerOzone.monthly.components.trend <-
  Kendall::MannKendall(GaringerOzone.monthly.components.ts)
# summary of non seasonal
summary(GaringerOzone.monthly.components.trend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The tau is -0.165 and the pvalue is 0.0075402. These are lower than the seasonal results, and provide stronger evidence that ozone concentrations have been decreasing over the past 10 years.