Assignment 7: Time Series Analysis

Megan McClaugherty

OVERVIEW

```
library(formatR)
library(knitr)
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=40), tidy=TRUE)
```

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

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 Getting working directory,
# installing/librarying packages, and
# setting theme.

getwd()
```

```
## [1] "/home/guest/EDA-Fall2022/Assignments"
```

```
library(tidyverse)
library(dplyr)
library(lubridate)
# install.packages('trend')
library(trend)
# install.packages('zoo')
library(zoo)
```

```
# install.packages('Kendall')
library(Kendall)
# install.packages('tseries')
library(tseries)
library(plyr)
library(readr)

A7Theme <- theme_light(base_size = 12) +
    theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
theme_set(A7Theme)

# 2 Importing the datasets in bulk into
# one dataframe called GaringerOzone
Ozonefiles = list.files(path = "/home/guest/EDA-Fall2022/Data/Raw/Ozone_TimeSeries",
    pattern = "*.csv", full.names = TRUE)


GaringerOzone = ldply(Ozonefiles, 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 Setting date column as date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date,
    format = "%m/%d/%Y")

# 4 Wrangling data set to only have the
# 3 columns. I had to rename the
# Daily...Concentration column because
# it loaded in with spaces instead of
# periods between the words.

colnames(GaringerOzone)[colnames(GaringerOzone) ==
    "Daily Max 8-hour Ozone Concentration"] <- "Daily.Max.8.hour.Ozone.Concentration"

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


# 5 Creating a new dataframe called
```

```
# Days with the specified dates and
# renaming the column to Date.
Days <- as.data.frame(seq(as.Date("2010-01-01"),
    as.Date("2019-12-31"), "days"))
colnames(Days)[colnames(Days) == "seq(as.Date(\"2010-01-01\"), as.Date(\"2019-12-31\"), \"days\")"] <-

# 6 Joining the Days dataframe with the
# Processed Garinger Ozone dataframe to
# a final GaringerOzone data frame with
# 3652 observations and 3 variables.

GaringerOzone <- left_join(Days, GaringerOzoneProcessed)
```

```
## Joining, by = "Date"
```

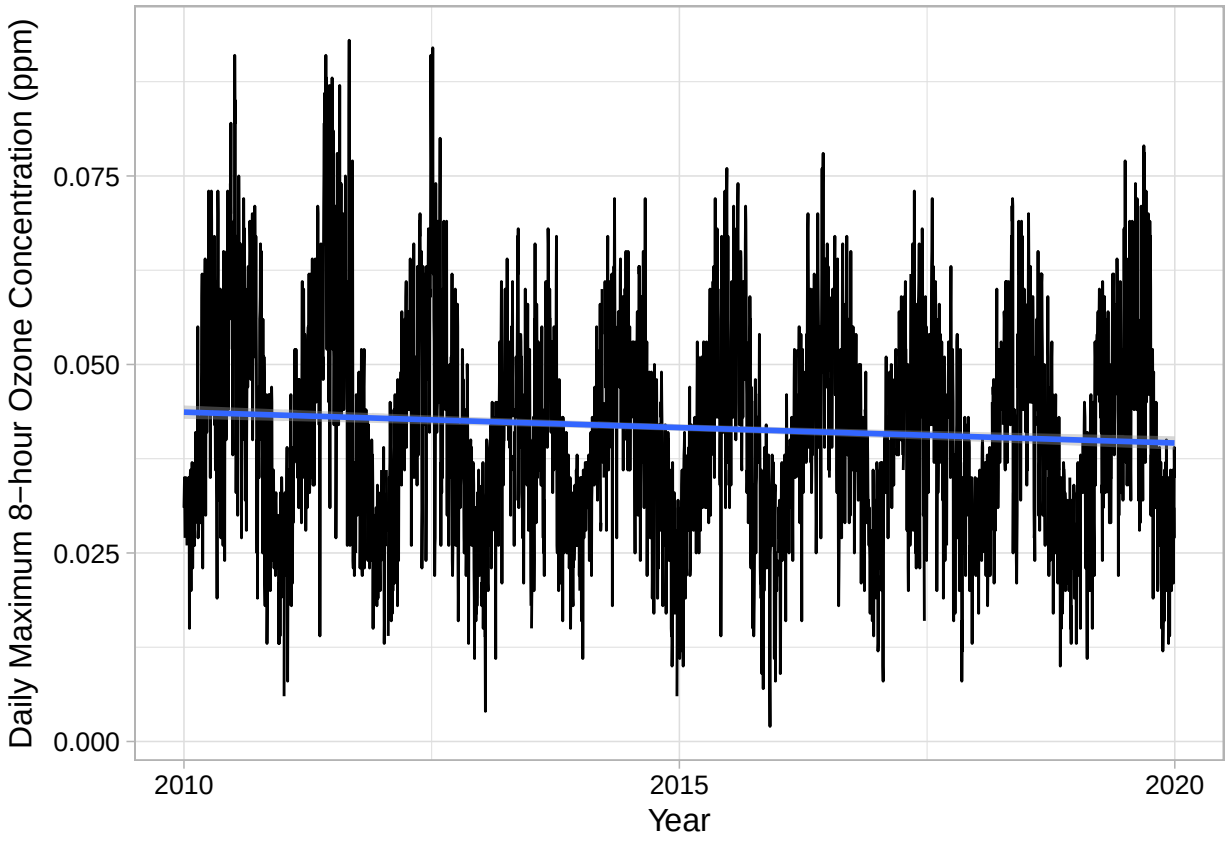
Visualize

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 Creating a line plot depicting
# ozone concentrations over time with a
# linear smooth line.

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() + geom_smooth(method = "lm") +
    labs(x = "Year", y = expression("Daily Maximum 8-hour Ozone Concentration (ppm)"))
```

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

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



Answer: The smoothed line is suggesting a negative trend over time.

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 for missing
# daily data.

GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co
summary(GaringerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                                                            NA's   :63
```

```
GaringerOzone$DAILY_AQI_VALUE <- na.approx(GaringerOzone$DAILY_AQI_VALUE)
```

Answer: The piecewise constant approach would interpolate the missing value to be equal to the measurement of the nearest date but since this is daily data with only single days missing, any

missing value would be equally distant to the measurement of the preceeding or following day. The spline interpolation approach uses a quadratic function to interpolate rather than a straight line. Since we were able to visualize a linear trend in our dataset, the linear interpolation makes sense to use.

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)

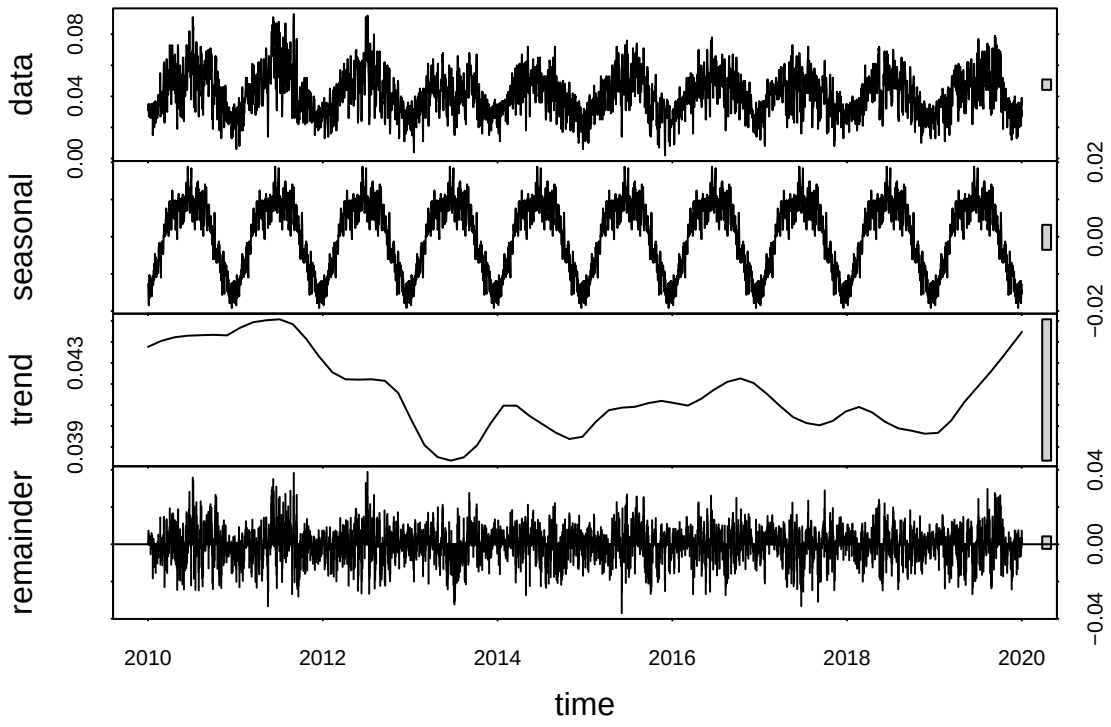
```
# 9 Creating the monthly average ozone
# concentrations dataframe.
GaringerOzone.monthly <- GaringerOzone %>%
    mutate(Year = year(Date)) %>%
    mutate(Month = month(Date)) %>%
    mutate(month_year = my(paste0(Month,
        "-", Year))) %>%
    select(month_year, Daily.Max.8.hour.Ozone.Concentration) %>%
    group_by(month_year) %>%
    dplyr::summarize(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration,
        n = n()))
```

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.

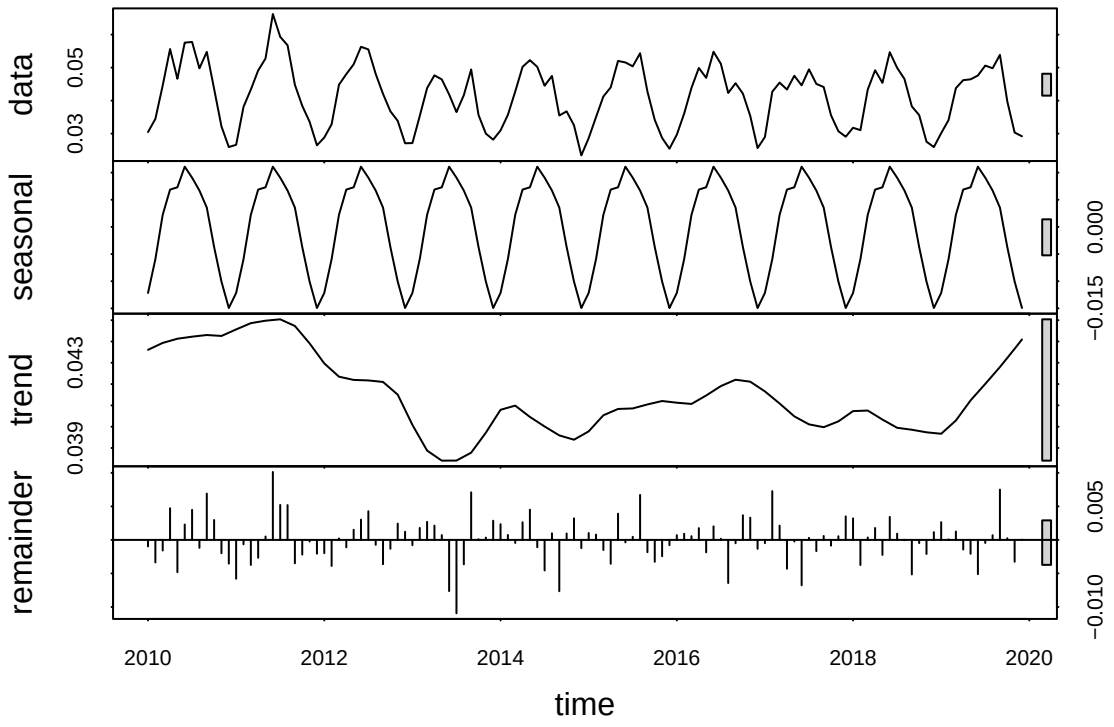
```
# 10 Generating the daily and monthly
# time series objects.

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
    start = c(2010), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone,
    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 Decomposing the daily and monthly
# time series to visualize the
# components.

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




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



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

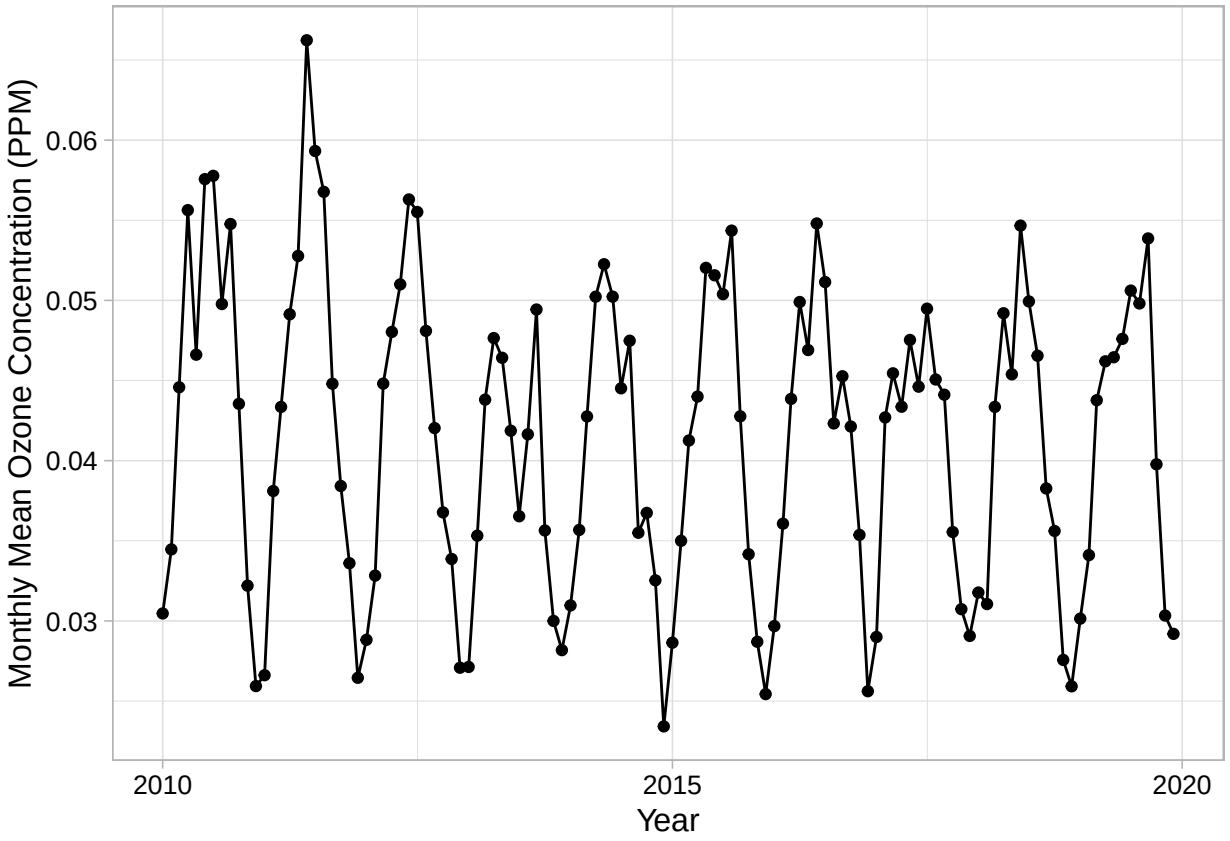
```
# 12 Using the seasonal Mann-Kendall
# trend analysis.

GaringerOzoneMonthlyTrend <- SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzoneMonthlyTrend)
```

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

Answer: By looking at the decomposition plot it is apparent that seasonality explains more of the variation in the ozone concentration data compared to the other possibilities.

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 Visualizing the montly average
# ozone concentration data.

GaringerMonthlyOzoneplot <- ggplot(GaringerOzone.monthly,
    aes(x = month_year, y = MeanOzone)) +
    geom_point() + geom_line() + xlab("Year") +
    ylab("Monthly Mean Ozone Concentration (PPM)")
print(GaringerMonthlyOzoneplot)
```



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 research question was has ozone concentration change during the 2010 to 2019 time period. Based on the two plots showing the ozone concentration over time as well as the monthly average concentration over time, there appears to be a slight negative trend. Based on the seasonal Mann-Kendall test, I would reject the null hypothesis that the ozone concentration is stationary over time, and that there is a trend in the seasonal component, however, this trend is very slight (p= 0.0467).

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
GaringerOzoneMonthNonSeas <- GaringerOzone.monthly.ts -
    GaringerOzoneMonthlyDecomp$time.series[,
        1]


# 16
GaringerMonthlyNonSeasTrend <- MannKendall(GaringerOzoneMonthNonSeas)
summary(GaringerMonthlyNonSeasTrend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
```

```
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The seasonal Man-Kendall had a p-value of 0.0467, so only a slight trend. After removing the seasonal component of the data the p-value was 0.0075. This suggests that the trend of decreasing ozone concentration over this time period is stronger without the seasonal component.