Assignment 8: Time Series Analysis

John Rooney

Spring 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
#check working directory
getwd()
```

```
## [1] "/Users/jrooney/Library/Mobile Documents/com~apple~CloudDocs/EDA_Sp23/EDA_Sp23"
```

```
#load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

```
library(zoo)
```

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

```
library(trend)
library(here)
```

```
## here() starts at /Users/jrooney/Library/Mobile Documents/com~apple~CloudDocs/EDA_Sp23/EDA_Sp23
```

```
library(Kendall)
library(forecast)
```

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

```
#set ggplot theme
my.theme <- theme_light(base_size = 14)+
  theme(axis.text = element_text(color = "grey19"),
        legend.position = "top",
        legend.justification = "left")
theme_set(my.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.

```
#2
#import datasets
raw.data = "Data/Raw/Ozone_TimeSeries"
Garinger2010<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFactors=T)
Garinger2011<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFactors=T)
```

```
Garinger2012<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFactors=T)
Garinger2013<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFactors=T)
Garinger2014<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFactors=T)
Garinger2015<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFactors=T)
Garinger2016<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFactors=T)
Garinger2017<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFactors=T)
Garinger2018<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFactors=T)
Garinger2019<-read.csv(
  here(raw.data, "EPAair_O3_GaringerNC2019_raw.csv"), stringsAsFactors=T)

#combine datasets
GaringerOzone <- rbind(Garinger2010, Garinger2011, Garinger2012, Garinger2013, Garinger2014, Garinger20
```

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
#set date column as a date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4
#wrangle dataset
GaringerOzone_wrangled <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_V

#5
#new days dataframe
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by="1 day"))
colnames(Days) <- "Date"

#6
#left join data frames
GaringerOzone <- left_join(Days, GaringerOzone_wrangled)
```

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

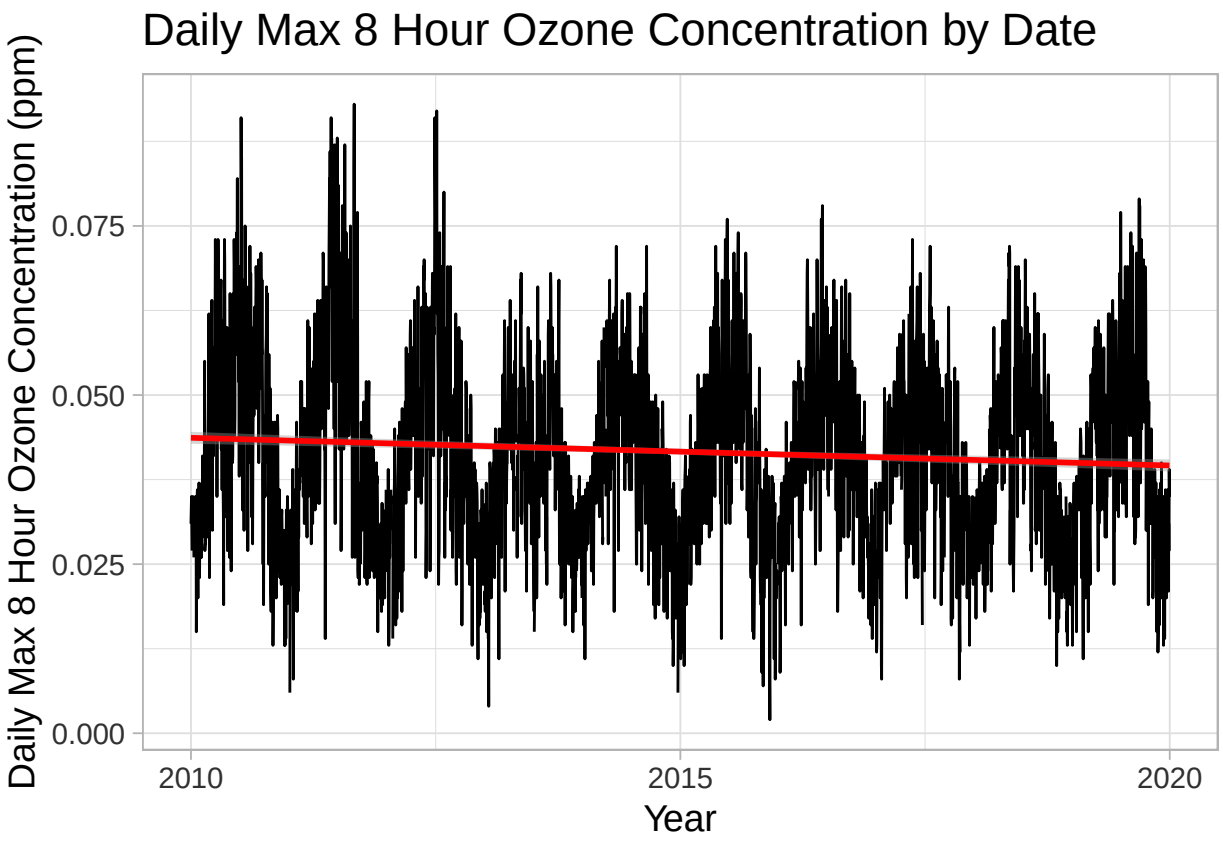
Visualize

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 concentration by date
GaringerOzonePlot <- ggplot(GaringerOzone,
                            aes(x=Date,
                                y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm, col="red") +
  labs(x="Year", y="Daily Max 8 Hour Ozone Concentration (ppm)", title="Daily Max 8 Hour Ozone Concentra
print(GaringerOzonePlot)
```

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

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



Answer: The plot does suggest a small negative trend over the course of the decade.

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
#use linear interpolation to fill in missing daily data for ozone
GaringerOzone <-
  GaringerOzone %>%
  mutate(Ozone_Concentration_clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Piecewise constant would have created data that didn't follow any trend in the data due to the assumption that missing data are equal in value that the nearest measurement. Spline interpolation would have had similar issues due to it's more complicated quadratic function rather than simply drawing a straight line.

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 new monthly data frame
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(mean(Ozone_Concentration_clean)) %>%
  mutate(Date =ym(paste0(Year,"-",Month)))
```

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

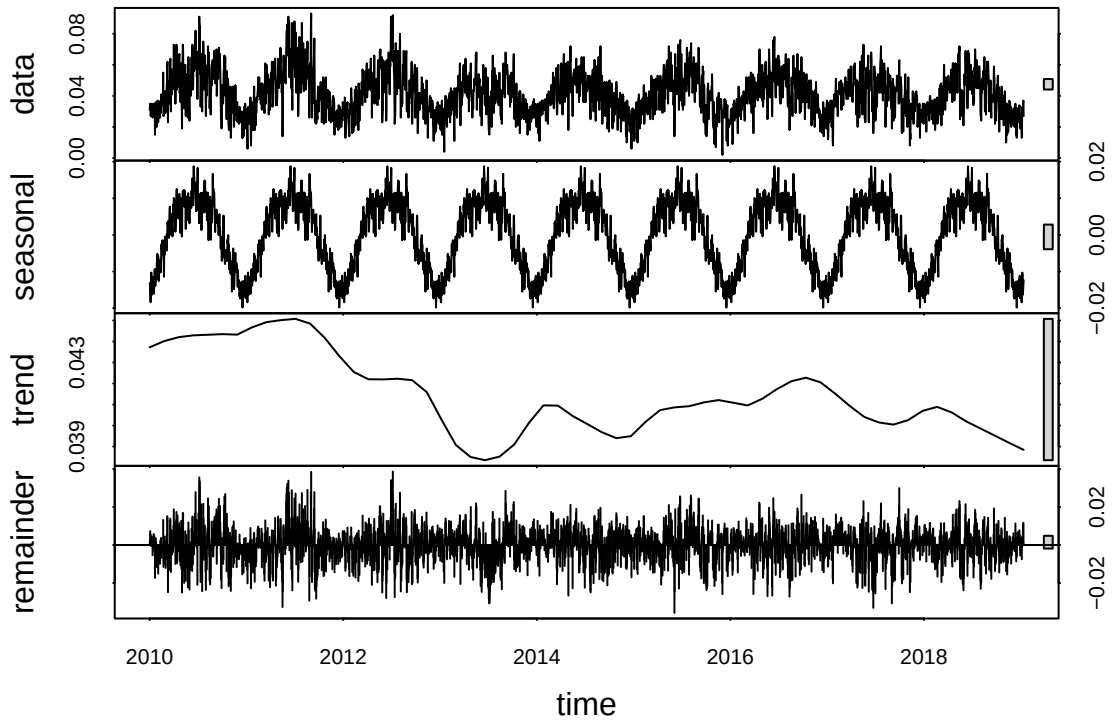
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10
#make time series of daily
start_month_daily <- month(first(GaringerOzone$Date))
start_year_daily <- year(first(GaringerOzone$Date))

end_month_daily <- month(last(GaringerOzone$Date))
end_year_daily <- year(last(GaringerOzone$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone[,4],
                             start=c(start_year_daily,start_month_daily),
                             end=c(end_year_daily, end_month_daily),
                             frequency = 365)

#make time series of monthly
start_month_monthly <- month(first(GaringerOzone.monthly$Date))
start_year_monthly <- year(first(GaringerOzone.monthly$Date))
```

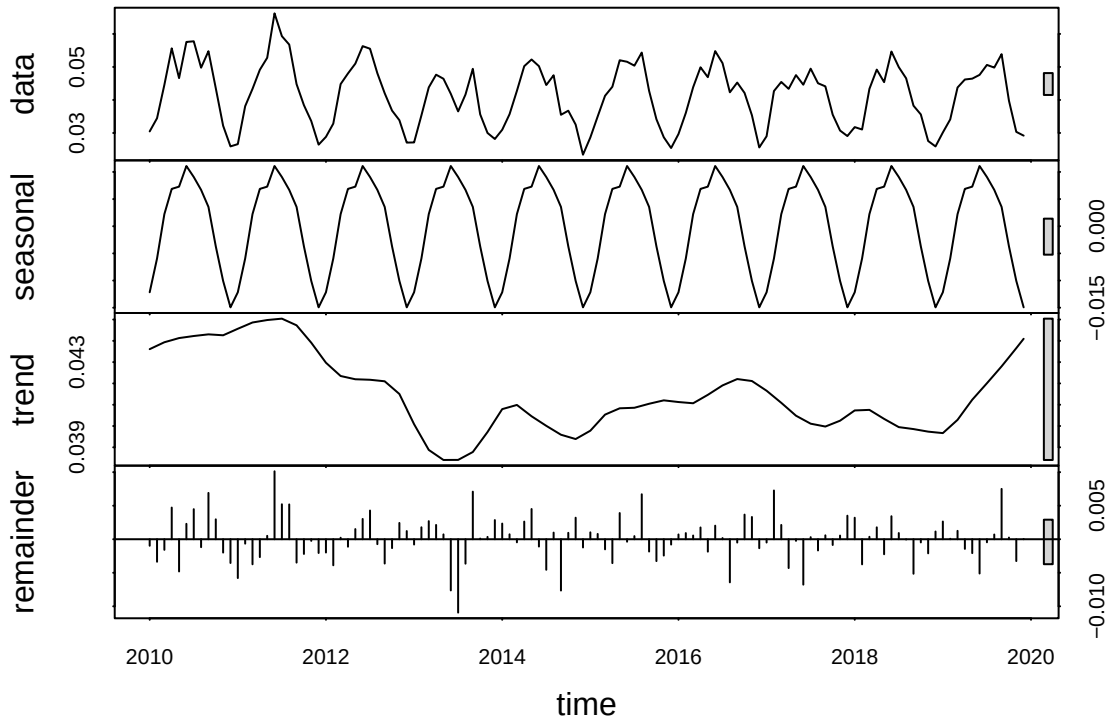
```
end_month_monthly <- month(last(GaringerOzone.monthly$Date))
end_year_monthly <- year(last(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$`mean(Ozone_Concentration_clean)`,
                               start=c(start_year_monthly,start_month_monthly),
                               end=c(end_year_monthly,end_month_monthly),
                               frequency=12)
```

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

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




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



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

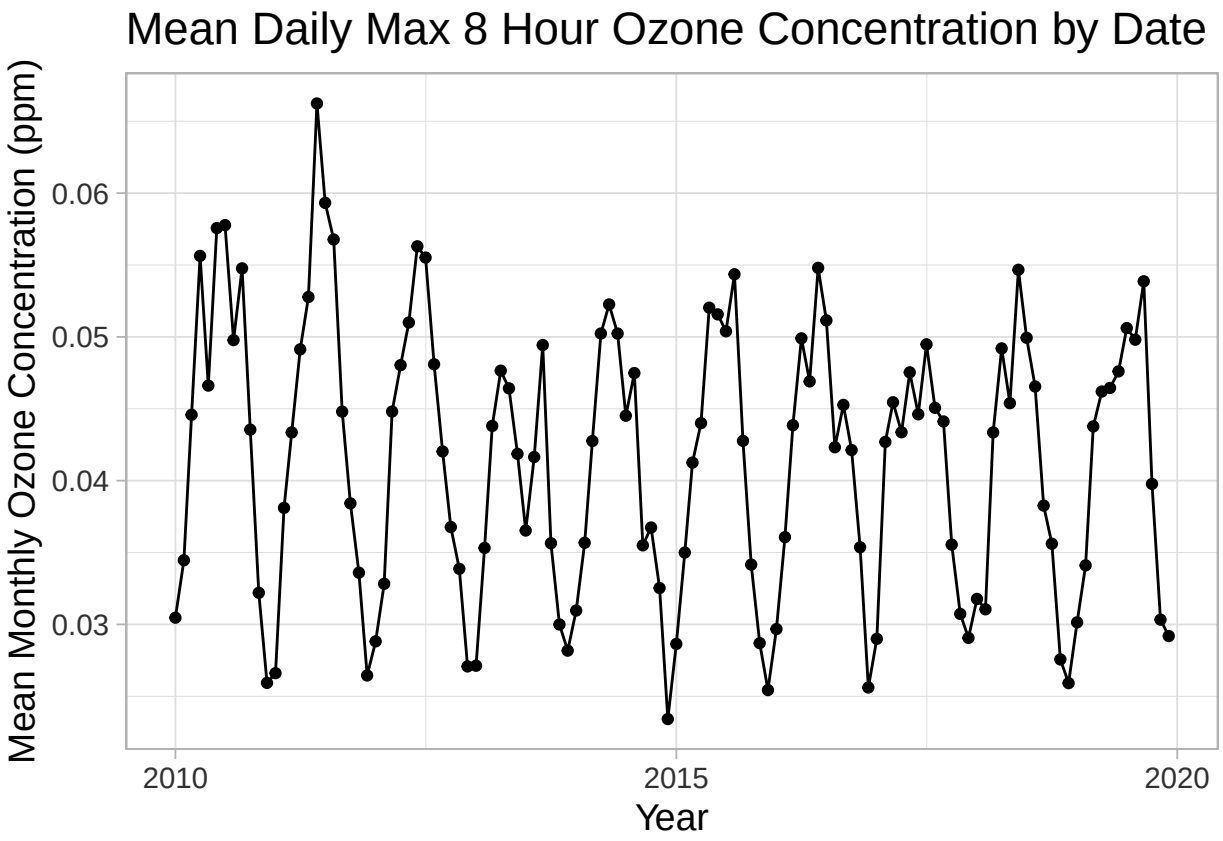
```
#12
GaringerOzone.monthly.ts.SMK <- SeasonalMannKendall(GaringerOzone.monthly.ts)
print(GaringerOzone.monthly.ts.SMK)
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The Seasonal Mann-Kendall is the most appropriate test as the time series still has seasonality in it. The normal Mann-Kendall test couldn't handle the seasonal data.

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(GaringerOzone.monthly,aes(x=Date,
                                 y=`mean(Ozone_Concentration_clean)`)) +
  geom_point() +
  geom_line() +
  labs(x="Year", y="Mean Monthly Ozone Concentration (ppm)", title="Mean Daily Max 8 Hour Ozone Concentr
```



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: Our research question, whether ozone concentration has changed over time at this particular research station, can be answered in the affirmative. The Seasonal Mann Kendall test we ran, which tests whether there is monotonic trend in a series, returned a p-value which can give us confidence in rejecting the null hypothesis that no trend exists in the series (p-value = 0.047).

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
deseasoned.GaringerOzone.monthly <- seasadj(GaringerOzone.monthly.ts.decomp)


#16

GaringerOzone.monthly.MK <- MannKendall(deseasoned.GaringerOzone.monthly)
print(GaringerOzone.monthly.MK)
```

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

Answer: The p-value for the Mann-Kendall is much lower than for the seasonal Mann-Kendall, with a p-value = 0.008.