Assignment 7: Time Series Analysis

Jackie Fahrenholz

OVERVIEW

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 Monday, March 14 at 7:00 pm.

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 wd
setwd("C:/Users/Jackie/Desktop/ENV872/Environmental_Data_Analytics_2022/Assignments")
getwd()
```

```
## [1] "C:/Users/Jackie/Desktop/ENV872/Environmental_Data_Analytics_2022/Assignments"
```

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- 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)
#build ggplot theme
mytheme <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right") +
  theme(plot.title = element_text(hjust = 0.5))
#set it as my 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
NC2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = T
NC2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = T
NC2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = T
NC2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = T
NC2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = T
NC2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = T
NC2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = T
NC2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = T
NC2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = T
NC2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = T
GaringerOzone <- rbind(NC2010, NC2011, NC2012, NC2013, NC2014, NC2015, NC2016, NC2017, NC2018, NC2019)
```

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 the date as a date
```

```
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
#check to make sure
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
# 4
#wrangle the dataset
GaringerOzone.wrangle <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
#generate df with dates from specific dates given
Days <- as.data.frame(seq(from = as.Date('2010-01-01'), to = as.Date('2019-12-31'), by = 1))

#rename the column
colnames(Days) <- c("Date")
#take a look
head(Days)
```

```
##         Date
## 1 2010-01-01
## 2 2010-01-02
## 3 2010-01-03
## 4 2010-01-04
## 5 2010-01-05
## 6 2010-01-06
```

```
# 6
#join the two data frames; dim 3652 rows and 3 columns
GaringerOzone <- left_join(Days, GaringerOzone.wrangle)
```

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

```
#take a look again
head(GaringerOzone)
```

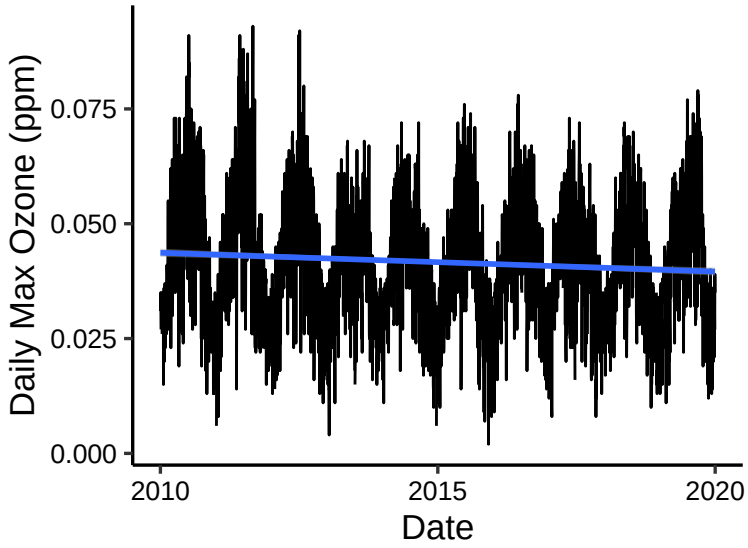
```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                   NA              NA
```

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 ppm over time
ggplot(GaringerOzone, aes(x = Date, y= Daily.Max.8.hour.Ozone.Concentration))+
  geom_line() +
  geom_smooth(method = "lm") +
  ylab("Daily Max Ozone (ppm)")
```

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



Answer: From this, our trend lines to have a decreasing slope, suggesting a negative trend in ozone over time; meaning ozone has decreased from 2010 to 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
#check for NAs
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   :0.04163                      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                           NA's   :63
```

```
#now that we know there are, fill the missing data
GaringerOzone_fill <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Linear interpolation draws points between the data to connect the dots. Based on our data and it having a similar pattern over time, this makes the most sense compared to the other interpolation types. Piecewise wouldn't work likely because we expect each day to have a different value, where this function would assume that missing days are the same as their "nearest neighbor". The spline method would use a quadratic function to determine the value of the nearest point, but because we have a trendline in this dataset that isn't necessary.

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)

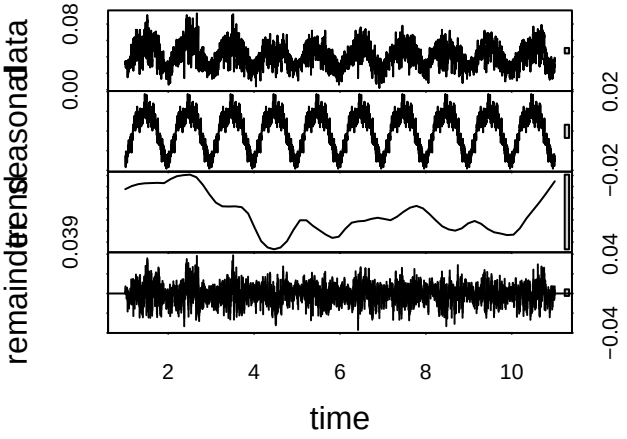
```
#9
#create a new data frame based on qualifications
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(GaringerOzone_fill, Month = format(Date,"%m")) %>%
  mutate(GaringerOzone_fill, Year = format(Date, "%Y")) %>%
  mutate(GaringerOzone_fill, Date = dmy(paste0("01-",Month,"-",Year))) %>%
  group_by(Date) %>%
  summarise(AvgOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

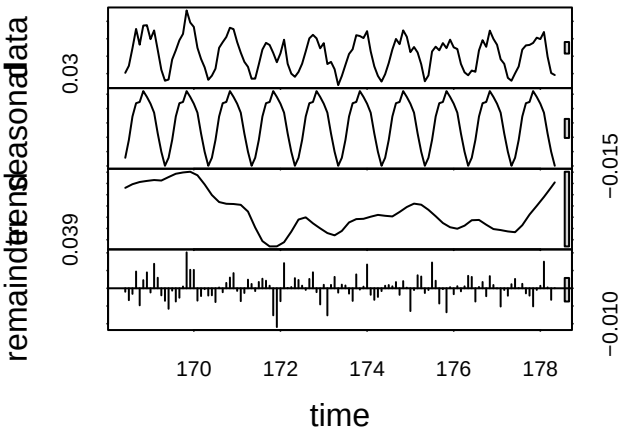
```
#10
#first set your month and year
fday <- day(first(GaringerOzone$Date))
fmonth <- month(first(GaringerOzone$Date))
fyear <- year(first(GaringerOzone$Date))
#create time series
GaringerOzone.daily.ts <- ts(GaringerOzone_fill$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(fday, fmonth, fyear),
                             frequency = 365)
#first month and year
frstmonth <- month(first(GaringerOzone.monthly$Date))
frstyear <- year(first(GaringerOzone.monthly$Date))
#create time series
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$AvgOzone,
                               start = c(frstmonth, frstyear),
                               frequency = 12)
```

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

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




```
#monthly decomp
month_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(month_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
#Monotonic trend; Mann-Kendall
monthly_ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly_ozone_trend1
```

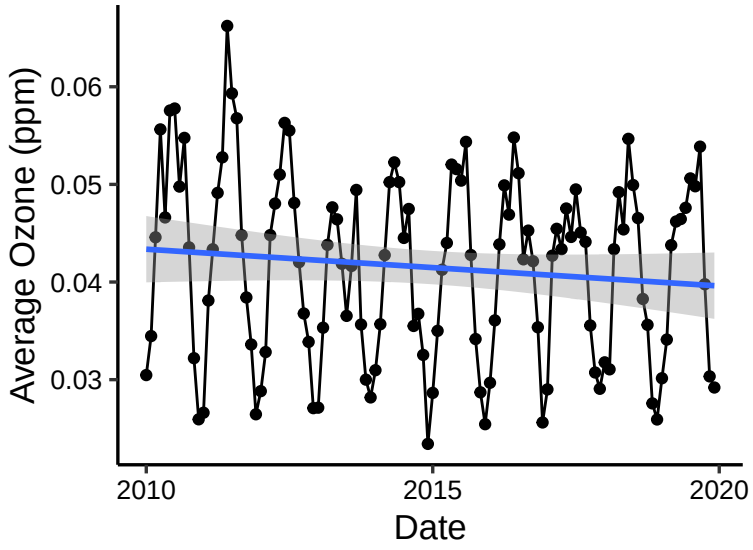
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: This test is appropriate based we know that our data has seasonality throughout the year and this test allows for seasonality, compared to the Mann-Kendall which doens't allow for seasonality. We also know our data is non-parametric meaning the data doesn't follow a specific distribution, so this test works well.

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
#visualize the data
ggplot(GaringerOzone.monthly, aes(x = Date, y = AvgOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Ozone (ppm)") +
  geom_smooth(method = lm)
```

```
## `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: Yes, there has been a change in ozone during the 2010s at this station. Over time we see a general decreasing trend seen in our visualization graphs. Because our data has seasonality, further analysis was conducted to prove that in fact there is a negative trend over time. Linear interpolation was conducted to fill in missing data points for daily values, used further to gather monthly averages across the 2010s. A `Seasonal Mann-Kendall` test was preformed, with results of a `p-vale = 0.046724` leading us to reject our null hypothesis; there is a trend present. A tau value of -0.143 means that we do in fact have a negative trend.

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
#subtract the seasonal component
GaringerOzone.monthly.ts_comp <- as.data.frame(month_decomp$time.series)

GaringerOzone.monthly.ts_comp <-
  GaringerOzone.monthly.ts_comp %>%
  mutate(Observed = GaringerOzone.monthly$AvgOzone, Date = GaringerOzone.monthly$Date)%>%
```

```
  mutate(NonSeasonal = Observed - seasonal)

#16
#run the Mann Kendall to compare
monthly_ozone_trend2 <- Kendall::MannKendall(GaringerOzone.monthly.ts_comp$NonSeasonal)

monthly_ozone_trend2
```

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

Answer: Even when we remove the non-seasonal data, we still find that there is a significant negative trend occurring in the data. The MannKendall test revealed a p-value of 0.0075402, leading us to reject our null hypothesis. We find here though that the slope has changed from -0.143 in the seasonal dataset to -0.165 in the non-seasonal.