Assignment 8: Time Series Analysis

Sam Campbell

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] "/home/guest/R/EDA-Spring2023"
```

```
#Load tidyverse, lubridate, zoo, and trend packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

```
install.packages("trend")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(trend)
install.packages("zoo")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
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/R/EDA-Spring2023
```

```
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x561102e64fa8>
## <environment: namespace:here>
```

```
#Set ggplot 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

#Import data sets
Garinger2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                         stringsAsFactors = TRUE)
Garinger2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                         stringsAsFactors = TRUE)

#Combine data sets into a single data frame
GaringerOzone <- rbind(Garinger2010, Garinger2011, Garinger2012, Garinger2013,
                       Garinger2014, Garinger2015, Garinger2016, Garinger2017,
                       Garinger2018, Garinger2019)
```

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

```
#Check class of date column
class(GaringerOzone$Date)
```

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

```
#4
#Wrangle dataset
GaringerOzoneWrangled <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5
#Generate daily dataset
Days <- as.data.frame(seq.Date(from=ymd("2010-01-01"),to=ymd("2019-12-31"),
                               by = "day"))

#Rename column name to Date
colnames(Days) <- c("Date")

#6
library(dplyr)

#Combine data frames
GaringerOzone <- left_join(Days, GaringerOzoneWrangled, by = "Date")
```

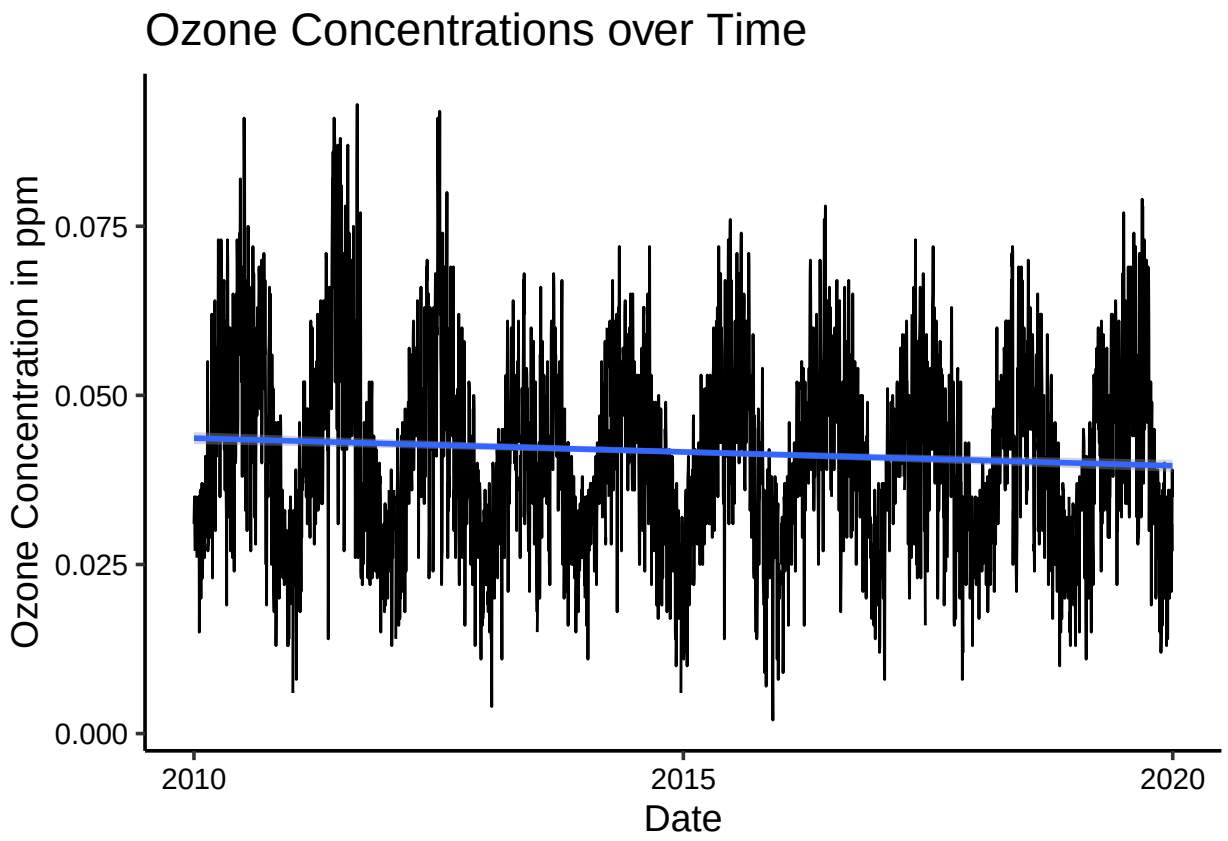
Visualize

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 over time
Plot7 <-
  ggplot(GaringerOzone,
       aes(x = Date,
           y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(title = "Ozone Concentrations over Time",
       x = "Date",
       y = "Ozone Concentration in ppm")
print(Plot7)
```

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

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



Answer: The plot suggests a slight decline in ozone concentrations 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
#Check missing values
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
```

```
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
```

```
#Add new column with no missing observations
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(Ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration),
         AQI.clean = zoo::na.approx(DAILY_AQI_VALUE))

#Select new cleaned up columns
GaringerOzone_cleanv2 <-
  select(GaringerOzone_clean, Date, Ozone.clean, AQI.clean)

#Check that NAs are gone
summary(GaringerOzone_cleanv2)
```

```
##       Date             Ozone.clean        AQI.clean
##  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   : 41.41
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100   3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300   Max.   :169.00
```

Answer: We used a linear interpolation because, as shown in the plot from question 7, the data is linear over time. Therefore, drawing straight lines between the known points is sufficient to estimate the position of the missing data points. A piecewise constant or spline interpolation could potentially generate more accurate results, but would require more complex computation that is not necessary in the case of this linear function

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 aggregated data frame by month
GaringerOzone.monthly <-
  GaringerOzone_cleanv2 %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  mutate(Date = paste(Year, Month, sep = " ")) %>%
  #ym(Date) %>%
  #arrange(Date) %>%
  #group_by(Date) %>%
  summarize(Mean_Ozone_Monthly = mean(Ozone.clean))
```

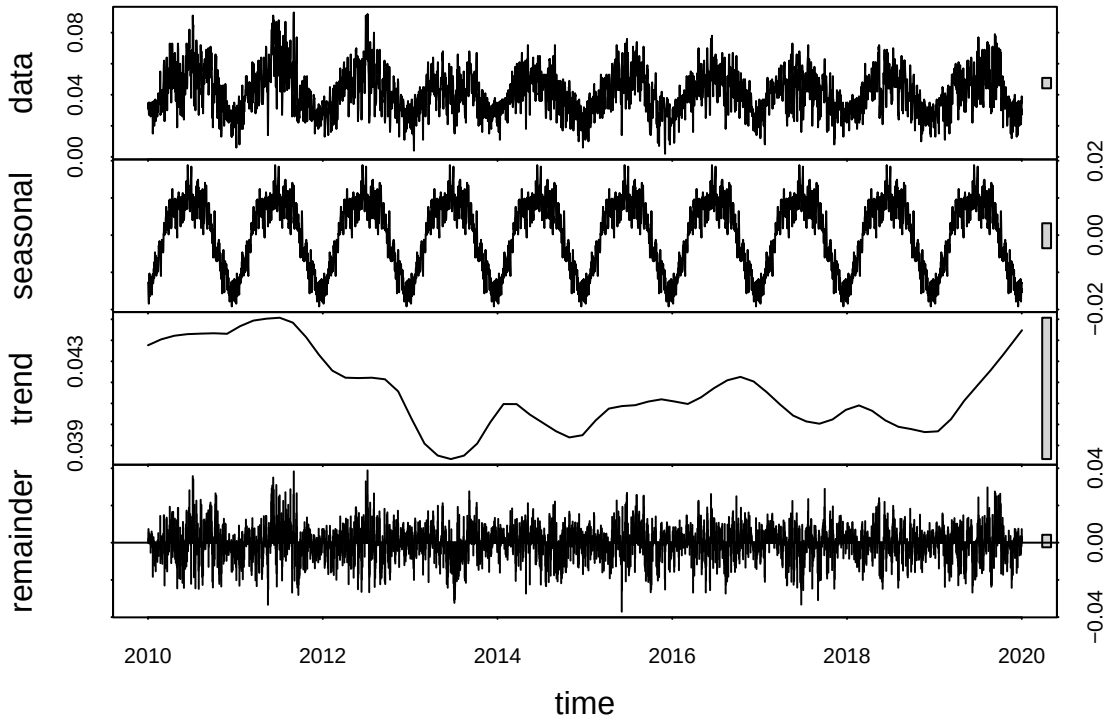
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10
#Generate daily time series object
GaringerOzone.daily.ts <- ts(GaringerOzone_cleanv2$Ozone.clean,
                             start = c(2010,1,1),
                             frequency = 365)

#Generate monthly time series object
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone_Monthly,
                               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 and plot daily time series object
Daily_decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(Daily_decomp)
```



```
#Decompose and plot monthly time series object
Monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
```

```
## Error in stl(GaringerOzone.monthly.ts, s.window = "periodic"): series is not periodic or has less tha
```

```
plot(Monthly_decomp)
```

```
## Error in plot(Monthly_decomp): object 'Monthly_decomp' not found
```

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 Mann-Kendall monotonic trend analysis for monthly Ozone series
Monthly_Ozone_smk <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

```
## Error in Kendall(1:length(x), x): length(x)<3
```

```
summary(Monthly_Ozone_smk)
```

```
## Error in summary(Monthly_Ozone_smk): object 'Monthly_Ozone_smk' not found
```

Answer: The seasonal Mann-Kendall is the most appropriate option because it is the only test we learned that can handle seasonality. As shown through the previous plots, our Ozone data varies seasonally.

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 mean monthly ozone over time
MeanMonthlyOzonePlot <-
  ggplot(GaringerOzone.monthly,
         aes(x = Date,
             y = Mean_Ozone_Monthly)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Ozone in ppm") +
  geom_smooth(method = "lm") +
print(MeanMonthlyOzonePlot)
```

```
## Error in print(MeanMonthlyOzonePlot): object 'MeanMonthlyOzonePlot' not found
```

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 results show that Ozone concentrations have changed over the course of the 2010s at the station. There has been an overall steady, albeit minor, decrease in Ozone concentrations over time, accounting for seasonal variations. The p value for the data is less than 0.05 (0.046724), which indicates that the downward trend in Ozone concentrations over the years is statistically significant.

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
#Extract components from decomposed monthly time series and turn them into data frames
GaringerOzone.monthly_Components <-
  as.data.frame(Monthly_decomp$time.series[,1:3])
```

```
## Error in as.data.frame(Monthly_decomp$time.series[, 1:3]): object 'Monthly_decomp' not found
```

```
GaringerOzone.monthly_Components <-
  mutate(GaringerOzone.monthly_Components,
         Observed = GaringerOzone.monthly$Mean_Ozone_Monthly,
         Date = GaringerOzone.monthly$Date)
```

```
## Error in mutate(GaringerOzone.monthly_Components, Observed = GaringerOzone.monthly$Mean_Ozone_Monthly
```

```
# Visualize how the trend maps onto the data
ggplot(GaringerOzone.monthly_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = trend, x = Date), color = "#c13d75ff") +
  geom_hline(yintercept = 0, lty = 2) +
  ylab("Mean Monthly Ozone in ppm")
```

```
## Error in ggplot(GaringerOzone.monthly_Components): object 'GaringerOzone.monthly_Components' not fou
```

```
# Visualize how the seasonal cycle maps onto the data
ggplot(GaringerOzone.monthly_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = seasonal, x = Date), color = "#c13d75ff") +
  geom_hline(yintercept = 0, lty = 2) +
  ylab("Mean Monthly Ozone in ppm")
```

```
## Error in ggplot(GaringerOzone.monthly_Components): object 'GaringerOzone.monthly_Components' not fou
```

```
#Turn non-seasonal data frame into a time series
GaringerOzone.monthly_Components.ts <-
  ts(GaringerOzone.monthly_Components$remainder,
     start = c(2010,1),
     frequency = 12)
```

```
## Error in is.data.frame(data): object 'GaringerOzone.monthly_Components' not found
```

```r
#16
#Run Mann Kendall on non-seasonal Ozone monthly series
Monthly_Ozone_mk <- Kendall::MannKendall(GaringerOzone.monthly_Components.ts)
```

```
## Error in Kendall(1:length(x), x): object 'GaringerOzone.monthly_Components.ts' not found
```

```r
summary(Monthly_Ozone_mk)
```

```
## Error in summary(Monthly_Ozone_mk): object 'Monthly_Ozone_mk' not found
```

Answer: When we run the Mann-Kendall test on the non-seasonal Ozone monthly series, the result is quite different from the Seasonal Mann-Kendall we previously ran on the complete series. Whereas the Seasonal Mann-Kendall on the complete series suggested a statistically signficant relationship between the change in the time and changes in Ozone levels with a p value less than 0.05, the Mann-Kendall on the non-seasonal series has a p value greater than 0.05 (0.80122), indicating that the change in Ozone concentrations over time for the non-seasonal series is not statistically significant.