Assignment 8: Time Series Analysis

Aditi Jackson

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

```
# checking working directory is ENV872 Setup (local)
getwd()
```

```
## [1] "/Users/aditijackson/ENV872 Setup (local)"
```

```
# installing packages
#install.packages("trend")
#install.packages("zoo")
#install.packages("Kendall")
#install.packages("tseries")

# 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 1.9.3     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(lubridate)
library(trend)
library(zoo)
```

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

```
library(Kendall)
library(tseries)
```

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

```
library(here)
```

```
## here() starts at /Users/aditijackson/ENV872 Setup (local)
```

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

# importing data
EPA_2010 <-
  read.csv(
    "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
    stringsAsFactors = TRUE)
EPA_2011 <-
  read.csv(
    "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
    stringsAsFactors = TRUE)
EPA_2012 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
  stringsAsFactors = TRUE)
EPA_2013 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
```

```
  stringsAsFactors = TRUE)
EPA_2014 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
  stringsAsFactors = TRUE)
EPA_2015 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
  stringsAsFactors = TRUE)
EPA_2016 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
  stringsAsFactors = TRUE)
EPA_2017 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
  stringsAsFactors = TRUE)
EPA_2018 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
  stringsAsFactors = TRUE)
EPA_2019 <- read.csv(
  "Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
  stringsAsFactors = TRUE)

# combing the datasets using rbind() function since
## all column names are the same
GaringerOzone <- rbind(EPA_2010,EPA_2011,EPA_2012,EPA_2013,EPA_2014,
                       EPA_2015,EPA_2016,EPA_2017,EPA_2018,EPA_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
# converting date column to date object
GaringerOzone$Date <- as.Date(GaringerOzone$Date,format="%m/%d/%Y")

# 4
# selecting columns to create abbreviated data set
GO_Sub <- select(GaringerOzone,
  Date,Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
# creating variables for start and end date
start_date <- as.Date("2010-01-01")
```

```r
end_date <- as.Date("2019-12-31")

# using objects to create a data frame of all dates between start and end date
Days <- as.data.frame(seq(start_date, end_date, by = 1))

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

# 6
# Using left join to combine Days and GaringerOzone_Subset data frames.
## Missing dates have been added and values populated with N/A
GO_Sub_Complete <- left_join(Days,GO_Sub)
```

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

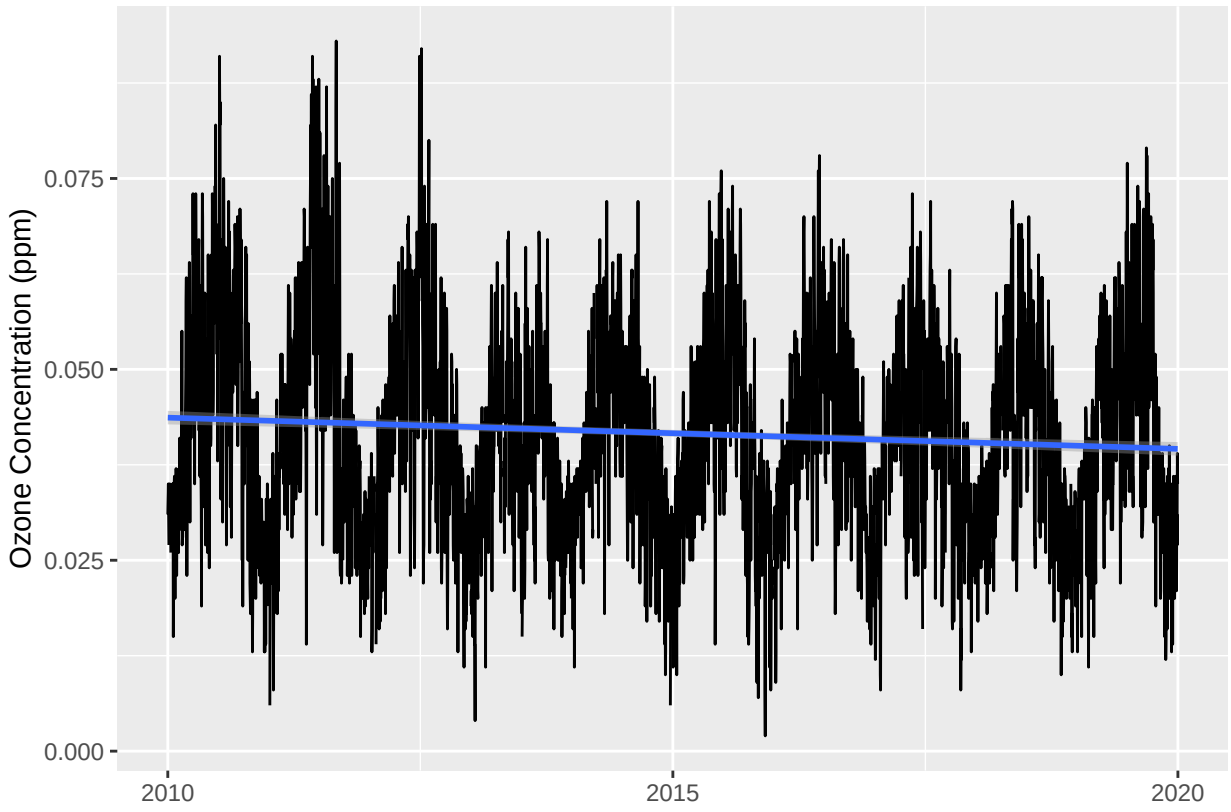
Visualize

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?

```r
#7
# plotting ozone concentrations over time
ggplot(GO_Sub_Complete,aes(
  x=Date,y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method="lm")+
  labs(x = "", y = "Ozone Concentration (ppm)")
```

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

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



Answer:The plot suggests a gradual decrease in Ozone concentrations over time. This can be seen by the slighly negative slope of the trend line.

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
summary(GO_Sub_Complete$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
# using a linear interpolation to fill in missing daily data for ozone concentration
GO_LinInterp <-
  GO_Sub_Complete %>%
  mutate(
    Daily.Max.8.hour.Ozone.Concentration =
      na.approx(Daily.Max.8.hour.Ozone.Concentration))

# checking that NAs in column were filled in
summary(GO_LinInterp$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: We used a linear interpolation and not spline or piecewise because the data fits a linear trend. For spline, the data would need to be quadratic and for piecewise the missing data would be assumed to be equal to its "nearest neighbor," which would not give us as good of a fit as using linear interpolation.

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
GaringerOzone.monthly <- GO_LinInterp %>%
  # creating separate month and year columns
  mutate(Year = year(Date),Month = month(Date)) %>%
  # grouping by Month and Year
  group_by(Year,Month) %>%
  # generating means for Ozone concentrations
  summarize(
    Mean_Mo_Ozone_Con =
      mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

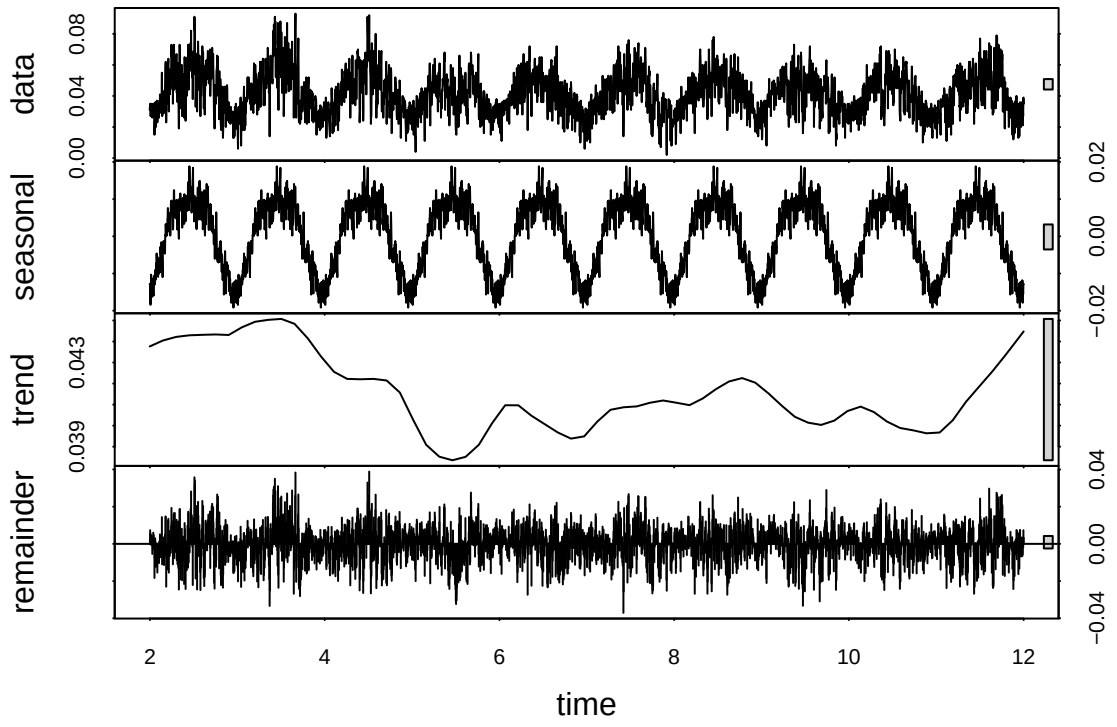
```
# creating new column in M-Y format
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = my(paste0(Month,"-",Year)))
```

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.

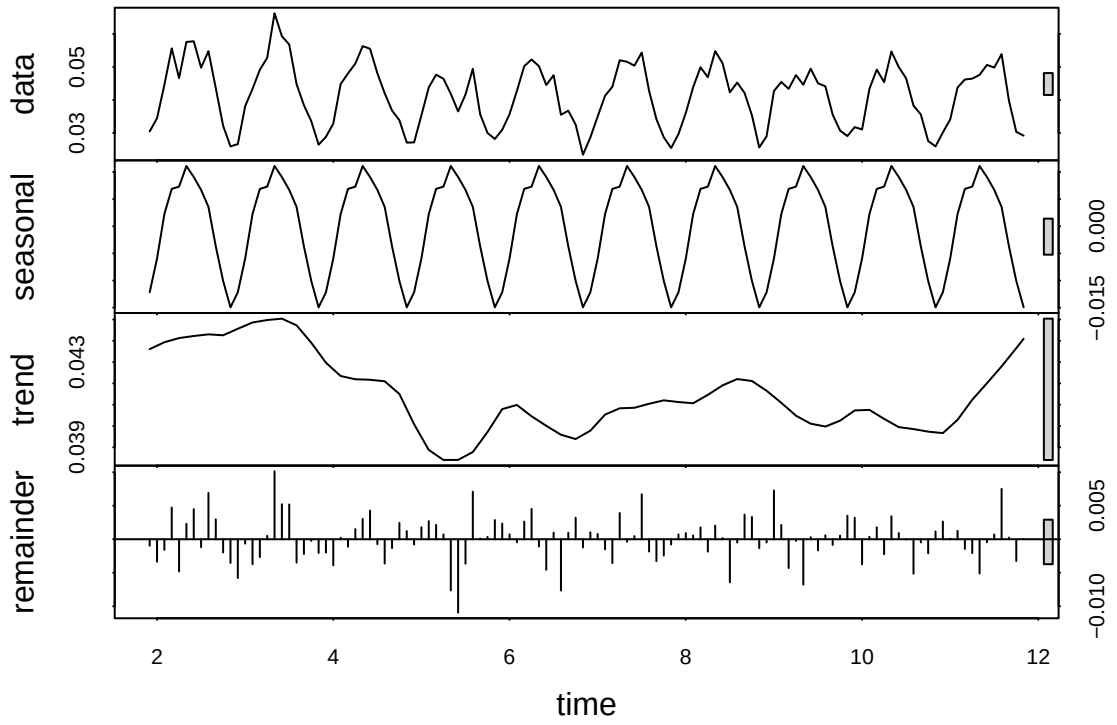
```
#10

# creating a time series object for Daily Ozone values
GaringerOzone.daily.ts <- ts(
  GO_LinInterp$Daily.Max.8.hour.Ozone.Concentration,start(2010,1),
          frequency = 365)

# creating a time series object for Monthly Ozone values
GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$Mean_Mo_Ozone_Con, start(2010,1),
          frequency = 12)
```

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

```r
#11
# decomposing daily time series object
GO_Daily_Decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")

# decomposing monthly time series object
GO_Monthly_Decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# plotting daily decomposition
plot(GO_Daily_Decomp)
```




```r
# plotting monthly decomposition
plot(GO_Monthly_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?

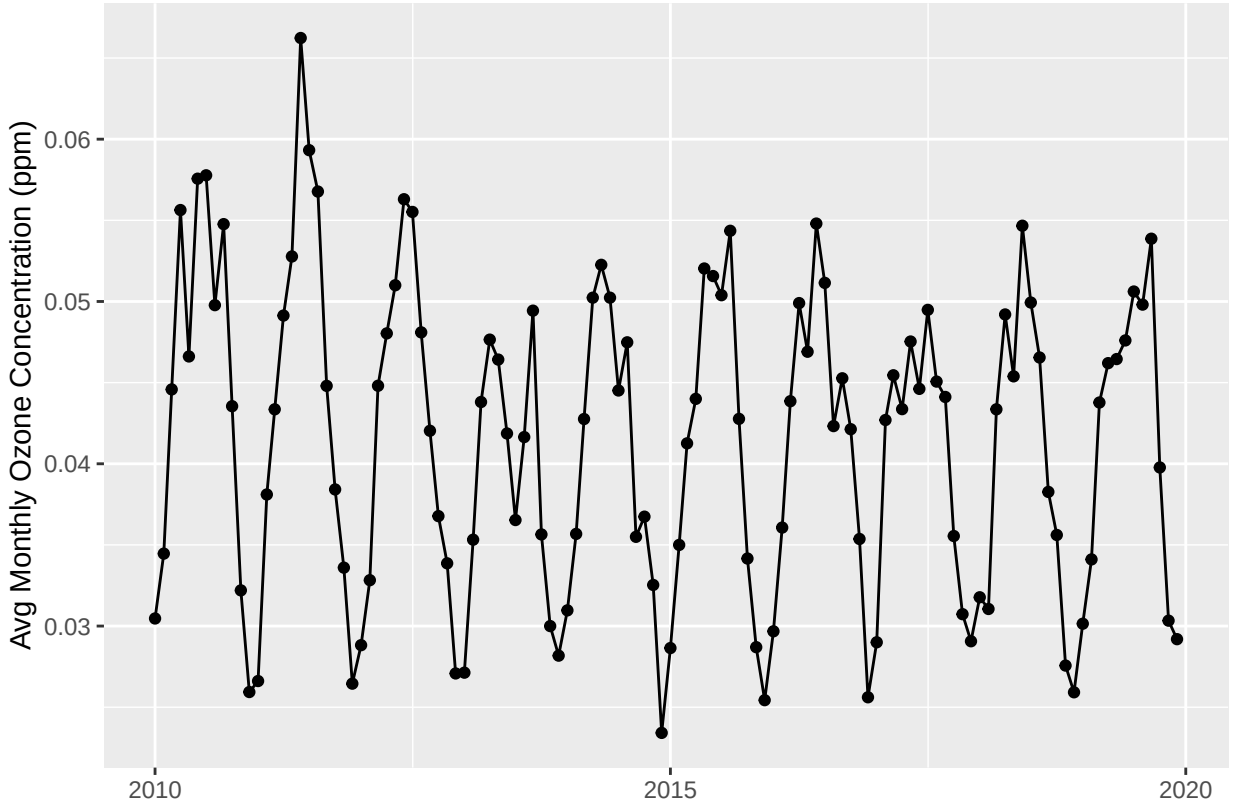
```
#12
# running Mann-Kendall test on monthly Ozone
GO_Data_MKT_Initial <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# displaying results of test
summary(GO_Data_MKT_Initial)
```

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

Answer: The seasonal Mann-Kendall test is most appropriate for the monthly Ozone data it's seasonal and non-parametric (no assumptions about the frequency distribution).

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
# plotting mean monthly ozone concentrations over time
ggplot(GaringerOzone.monthly,
       aes(x=Date,y=Mean_Mo_Ozone_Con))+
  geom_point()+
  geom_line()+
  labs(x="",y="Avg Monthly Ozone Concentration (ppm)")
```



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 asks if ozone concentrations changed over the 2010s at this station. Based on the graph, it appears that Ozone concencentrations are seasonal and have changed over time. The statistical test outcome validates this conclusion given that the p-value is less than alpha = 0.05, meaning we would reject the null hypothesis that there is not change in concentrations. The negative tau value tells us that the slope is negative and therefore ozone concentrations have decreased over time (2-sided pvalue = 0.046, 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
# separating out seasonal component of time series (Jon Joyner helped me figure out how to do question
GO_Seasonal_Monthly.ts <-
  (GO_Monthly_Decomp$time.series[,"seasonal"])

# subtracting seasonal component from original time series
GO_Unseasoned <- GaringerOzone.monthly.ts - GO_Seasonal_Monthly.ts

#16
# running Mann Kendall test
```

```r
GO_Data_MKT_Unseasoned <- Kendall::MannKendall(GO_Unseasoned)

# summarizing unseasonal data
summary(GO_Data_MKT_Unseasoned)
```

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

```r
#summarizing seasonal data
summary(GO_Data_MKT_Initial)
```

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

Answer: Comparing the tests, the p-value of the non-seasonal Mann-Kendall is lower than the p-value of the seasonal test. This makes logical sense because in the non-seasonal test, you are removing the variability due to seasonal changes, which strengths the relationship between Ozone and time horizon. The tau value shows a slightly steeper slope for nonseasonal, which verifies that removing the seasonal component strengthens the relationship between ozone and time.