Assignment 8: Time Series Analysis

Eric Newton

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

```
getwd()

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 1.9.2     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(zoo)
```

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

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

```
## here() starts at C:/Users/enewt/OneDrive/Documents/Duke/ENV872/EDE_Fall2023
```

```
mytheme <- theme_classic(base_size = 14)+
  theme(legend.background = element_rect(
    color = "grey",
    fill = "white"),
    plot.title = element_text(hjust = 0.5, size = 12),
    legend.position = "bottom")
theme_set(mytheme)

#set theme for the entire session using theme_set
```

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

GaringerOzone <- rbind(Garinger2010, Garinger2011, Garinger2012, Garinger2013, Garinger2014, Garinger201

#imported data sets individually and then combined them using rbind
```

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.

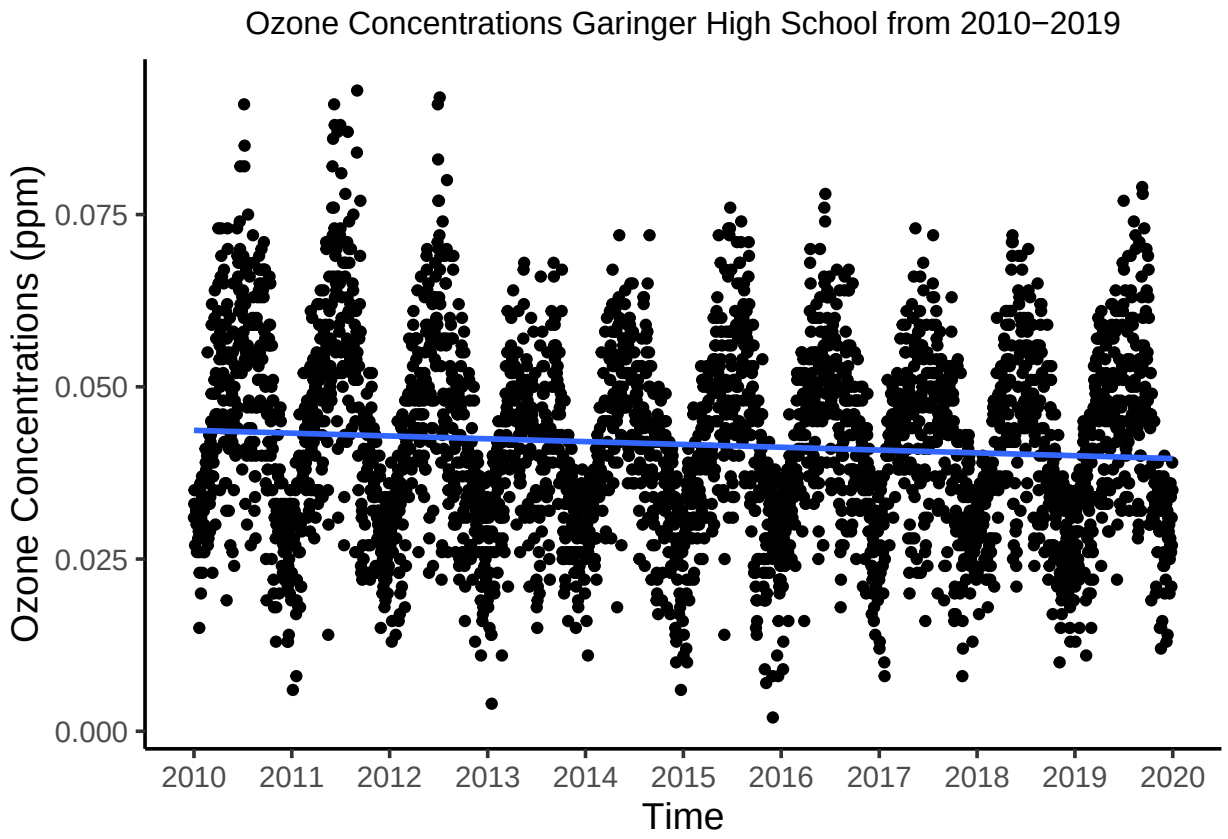
```
# 3
GaringerOzone$Date <- mdy(GaringerOzone$Date)


# 4
GO.Processed <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

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

# 6
GaringerOzone <- left_join(Days, GO.Processed, by = "Date")

#wrangled data to create dataframe with correct columns and data types. Then
#created a sectond dataframe with dates for the time series and merged this
#dataframe with the original to create a df suitable for a time series analysis
```

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
Ozone.over.time <-
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
geom_point() +
geom_smooth(method = "lm", se = FALSE) +
labs(
  x = "Time",
  y = "Ozone Concentrations (ppm)",
  title = "Ozone Concentrations Garinger High School from 2010-2019") +
scale_x_date(date_breaks = "1 year", date_labels = "%Y")
print(Ozone.over.time)
```

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

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

## Warning: Removed 63 rows containing missing values (`geom_point()`).
```




```
#created a line plot to visualize the dataset. Used geom_smooth with a linear
#model as the impot to show linear trend in the data
```

Answer: Yes, the plot suggests a slight decresing trend in ozone concentration 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
summary(GaringerOzone$DAILY_AQI_VALUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2.00   30.00   38.00   41.57   47.00  169.00      63
```

```
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(
    DAILY_AQI_VALUE.clean = zoo::na.approx(DAILY_AQI_VALUE)) %>%
  mutate(
```

```
    Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %
  select(
    Date, DAILY_AQI_VALUE.clean, Daily.Max.8.hour.Ozone.Concentration.clean)

summary(GaringerOzone_clean$DAILY_AQI_VALUE.clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   30.00   38.00   41.41   47.00  169.00
```

```
#used a linear interpolation to fill in missing data using the na.approx function
#and selected the new interpolated data into a 'clean' dataframe.
```

Answer: We used a linear interpolation to fill in the missing data because the data follows a linear rather than a quadratic rise and fall, and the data changes day to day so an assumption that the data would be the same as the nearest neighbor would not produce accurate results.

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 <-
  GaringerOzone_clean %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  mutate(Date = make_date(Year, Month, day = 1)) %>%
  group_by(Year, Month, Date) %>%
  summarize(Mean.Monthly.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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

```
#wrangled this interpolated data into a new monthly data set and summarized the
#data into monthly mean values.
```

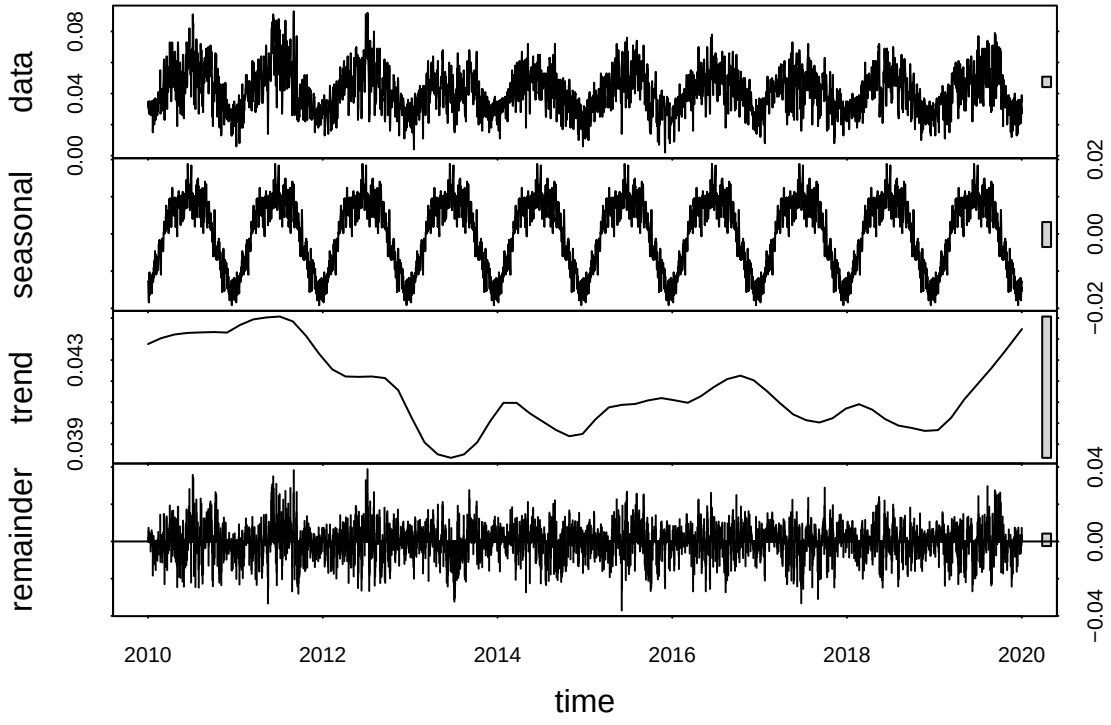
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10
GO_day <- day(first(GaringerOzone_clean$Date))
GO_month <- month(first(GaringerOzone_clean$Date))
GO_year <- year(first(GaringerOzone_clean$Date))
GaringerOzone.daily.ts <- ts(
  GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration.clean,
  start = c(GO_year, GO_month, GO_day),
  frequency=365)

Garinger_month <- first(GaringerOzone.monthly$Month)
```

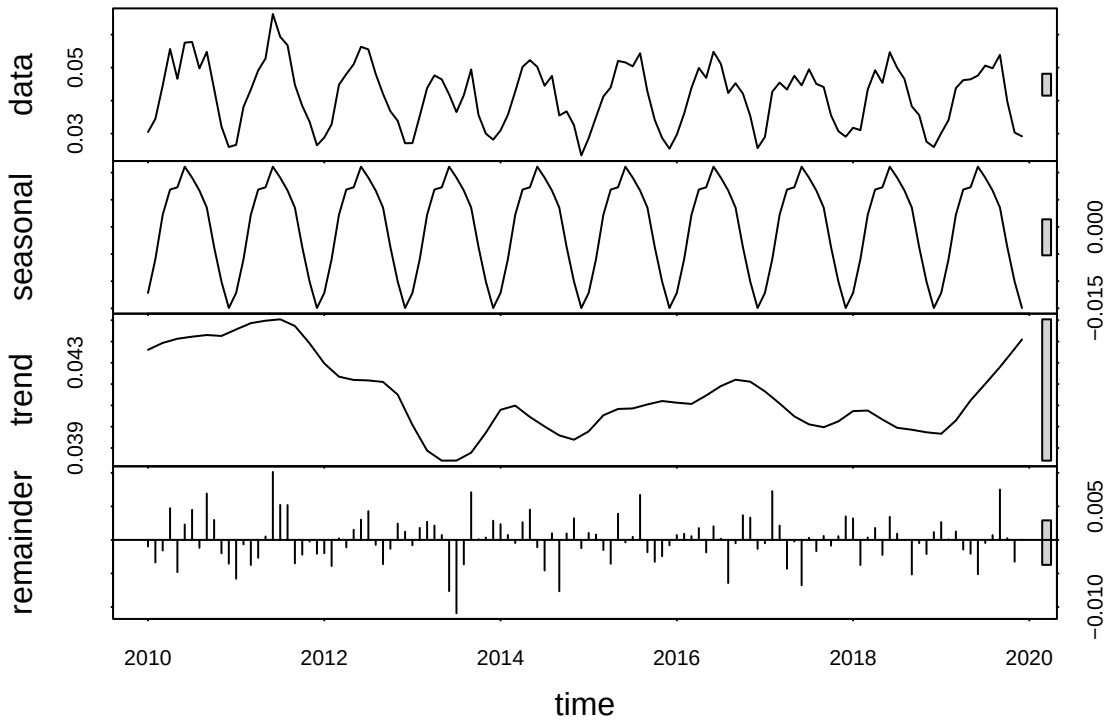
```
Garinger_year <- first(GaringerOzone.monthly$Year)
GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$Mean.Monthly.Ozone.Concentration,
  start = c(Garinger_year, Garinger_month),
  frequency=12)

#Generated time series objects for the daily observation and monthly average
#datasets.
```

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

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




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




```
#decomoposed the data using the stl function and plotted the decomposition to
#show seasonal, trend, and remainder plots.
```

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
Trend.monthly.ozone <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
print(Trend.monthly.ozone)
```

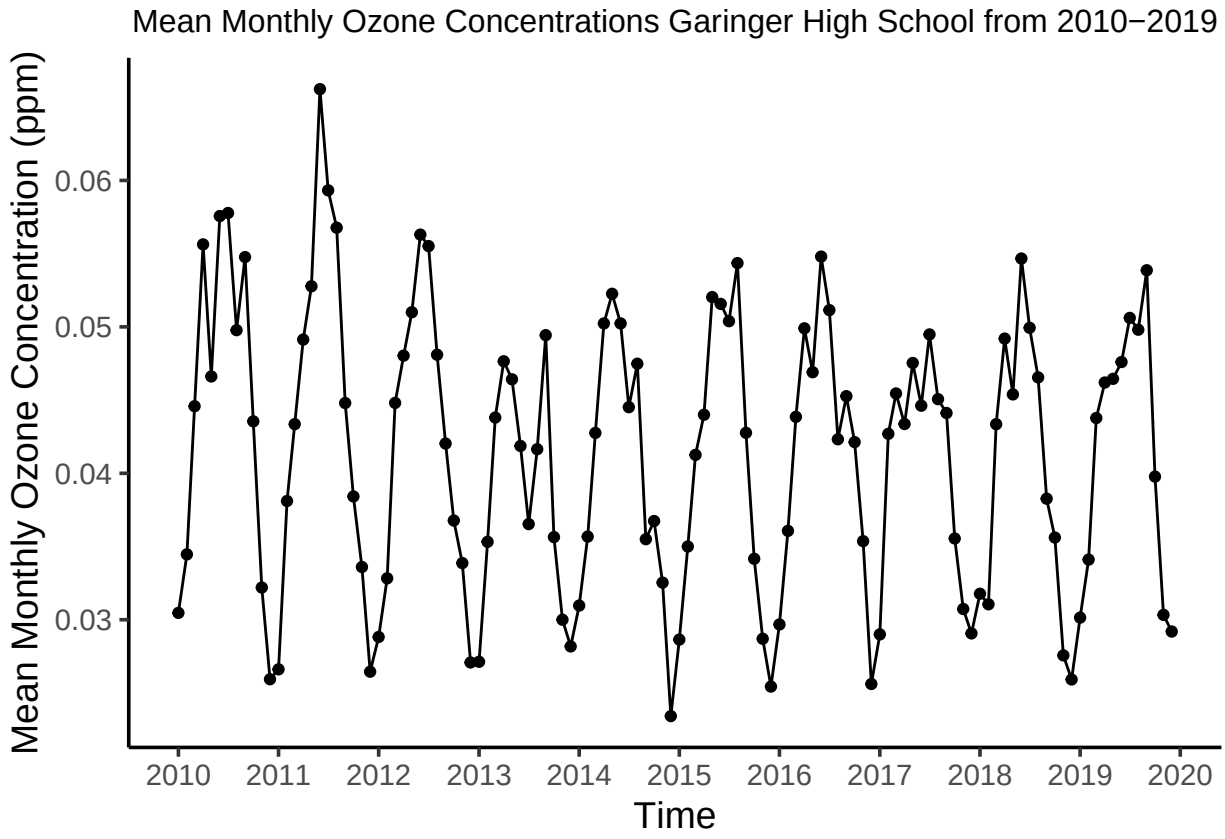
```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
#ran a trend analysis using the seasonal Mann-Kendall function
```

Answer: Seasonal Mann-Kendall is most appropriate beacuase the data is seasonal (it rises and falls in a yearly cycle), the data is non-parametric, and there is no missing 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.

```r
# 13
Mean.Monthly.Plot <-
  ggplot(GaringerOzone.monthly, aes(
    x = Date,
    y = Mean.Monthly.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
  labs(
  x = "Time",
  y = "Mean Monthly Ozone Concentration (ppm)",
  title = "Mean Monthly Ozone Concentrations Garinger High School from 2010-2019") +
scale_x_date(date_breaks = "1 year", date_labels = "%Y")
print(Mean.Monthly.Plot)
```




```r
#created a scatter plot of mean montly ozone concentration over time
```

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 demonstrate a clear seasonal pattern with ozone concentrations at the site rising in spring/summer and falling in fall/winter. There is a trend, albeit not very strong, between mean monthly ozone concentration and time (tau = -0.143, 2-sided pvalue =0.046724). This is most apparent by the height of the summer peak in ozone concentration falling throughout the 2010s.

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
Garinger.Components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

Garinger.Nonseasonal <- Garinger.Components - Garinger.Components$seasonal


#16
Garinger.Nonseasonal.ts <-
  ts(Garinger.Nonseasonal,
  start = c(Garinger_year, Garinger_month),
  frequency = 12)

Trend2.monthly.ozone <-
  Kendall::MannKendall(Garinger.Nonseasonal.ts)
print(Trend2.monthly.ozone)
```

```
## tau = -0.0275, 2-sided pvalue =0.45485
```

```
#subtracted out the seasonality and ran a trend analysis using the
#Mann-Kendall function
```

Answer: The trend analysis when seasonality is subtracted out shows that mean monthly ozone concentration is independent of time (tau = -0.0275, 2-sided pvalue =0.45485). This is unsuprising given the clear seasonality in the original series.