Assignment 8: Time Series Analysis

Kamil Burak Karayel

Spring 2025

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
```

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

```
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
```

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

```
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2025_kbk
```

```
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
library(ggthemes)
#create my theme
mytheme_kbk <- theme_base(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

#set my theme as default
theme_set(mytheme_kbk)
```

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
#import ten different datasets individually
EPAair_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFa

EPAair_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFa

EPAair_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFa
```

```
EPAair_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFa

EPAair_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFa

EPAair_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFa

EPAair_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFa

EPAair_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFa

EPAair_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFa

EPAair_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"), stringsAsFa

#merge ten datasets into one
GaringerOzone <- bind_rows(EPAair_2010,EPAair_2011,EPAair_2012,EPAair_2013,EPAair_2014,EPAair_2015,EPAa

#dimensions of this new dataset are 3589 observations of 20 variables
dim(GaringerOzone)
```

```
## [1] 3589   20
```

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
#change class of date column to "date" as it was factor before
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
#class is seen as "date" now
class(GaringerOzone$Date)
```

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

```
# 4
#include only three variables
Q4_Garinger <- GaringerOzone %>% select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)
#this new dataset includes 3589 observations of 3 variables
str(Q4_Garinger)
```

```
## 'data.frame':    3589 obs. of  3 variables:
##  $ Date                                : Date, format: "2010-01-01" "2010-01-02" ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 0.033 0.035 0.032 0.032 0
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 31 32 30 30 28 ...
```

```
# 5
#create new dataset named "Days" that includes only date column from 01.01.2010 to 12.31.2019
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))

#rename column name to "Date" as it was "seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"),
Days <-
  Days %>%
  rename( Date = `seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day")`)


# 6
# join Days and dataframe from Q4 by Date column
GaringerOzone <- left_join(Days,Q4_Garinger,by = "Date")

#this new dataframe consists of 3652 observations of 3 variables
str(GaringerOzone)
```

```
## 'data.frame':    3652 obs. of  3 variables:
##  $ Date                                : Date, format: "2010-01-01" "2010-01-02" ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 NA 0.033 0.035 0.032 0.03
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 NA 31 32 30 30 ...
```

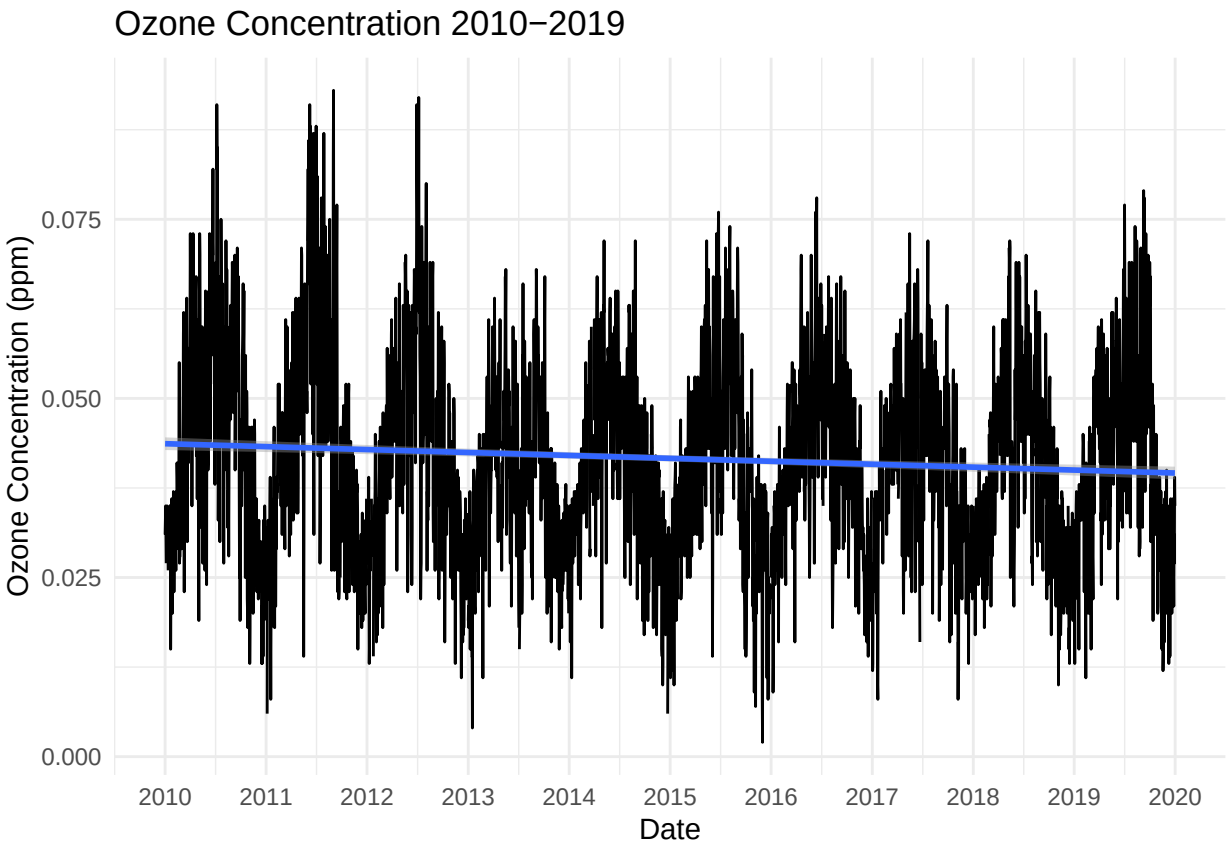
Visualize

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
#create a line plot
ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +  #add smoothed line
  labs(title = "Ozone Concentration 2010-2019",
    x="Date", y="Ozone Concentration (ppm)") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +  #add years to x scale
  theme_minimal()  #use minimal theme to add grid to the graph
```

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

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: There is a slightly decreasing trend over years. Also we can see seasonal changes in ppm levels.

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
#change missing values using linear interpolation with respect to previous and next cells
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Co
```

Answer: Linear interpolation fills in the missing values by a smooth increase od decrease in the gap. This doesn't change the average or the trend of the data points and compatible with the nature of the concentrations. Spline interpolation can cause oscillations and change the trend. Piecewise constant interpolation hold the last known value until the next data point and this would result instant jumps in data.

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)

```r
#9
#create two new columns "Year" and "Month" by using "Date" column
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year=format(Date, "%Y"),Month=format(Date,"%m"))

#merge "Year" and "Month" columns and create new column "Mean_Ozone" by taking mean of 8.hour.ozon.conc
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  group_by(Year,Month) %>%
  summarize(Mean_Ozone= mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

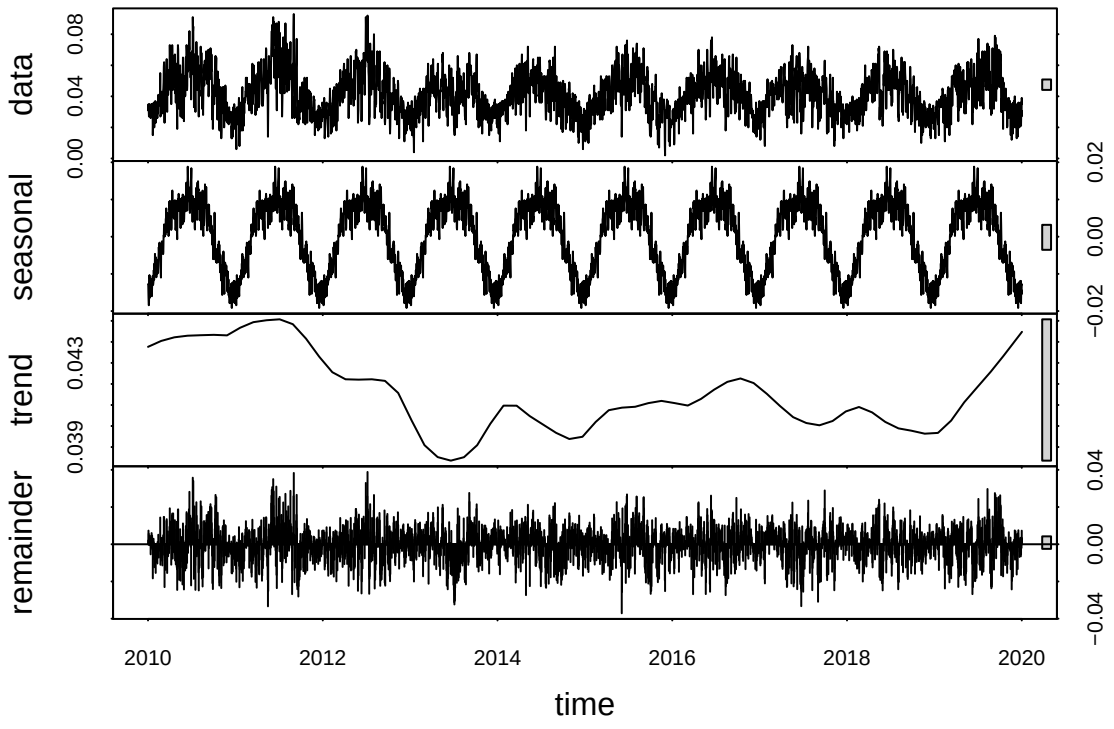
```r
#create new column "Date" by using "Year", "Month" columns and 01 for each combination.
GaringerOzone.monthly <- GaringerOzone.monthly %>%
 mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))
```

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.

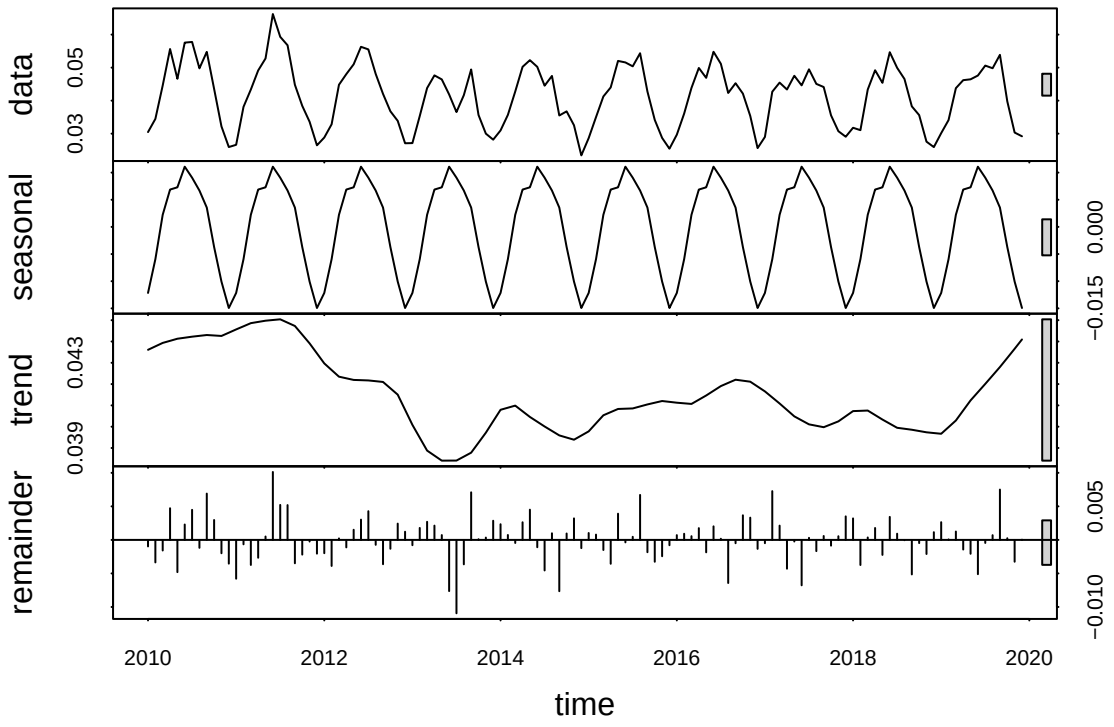
```r
#10
#create a ts object including daily ozone concentration from 2010-1-1 with 365 days cycle
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
 start = c(2010, 1, 1),
 frequency = 365)

#create a monthly ts including monthly averages
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
 start = c(2010, 1),
 frequency = 12)
```

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

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




```
#decompose monthly time series
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.decomposed)
```



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 test
Q12 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Q12
```

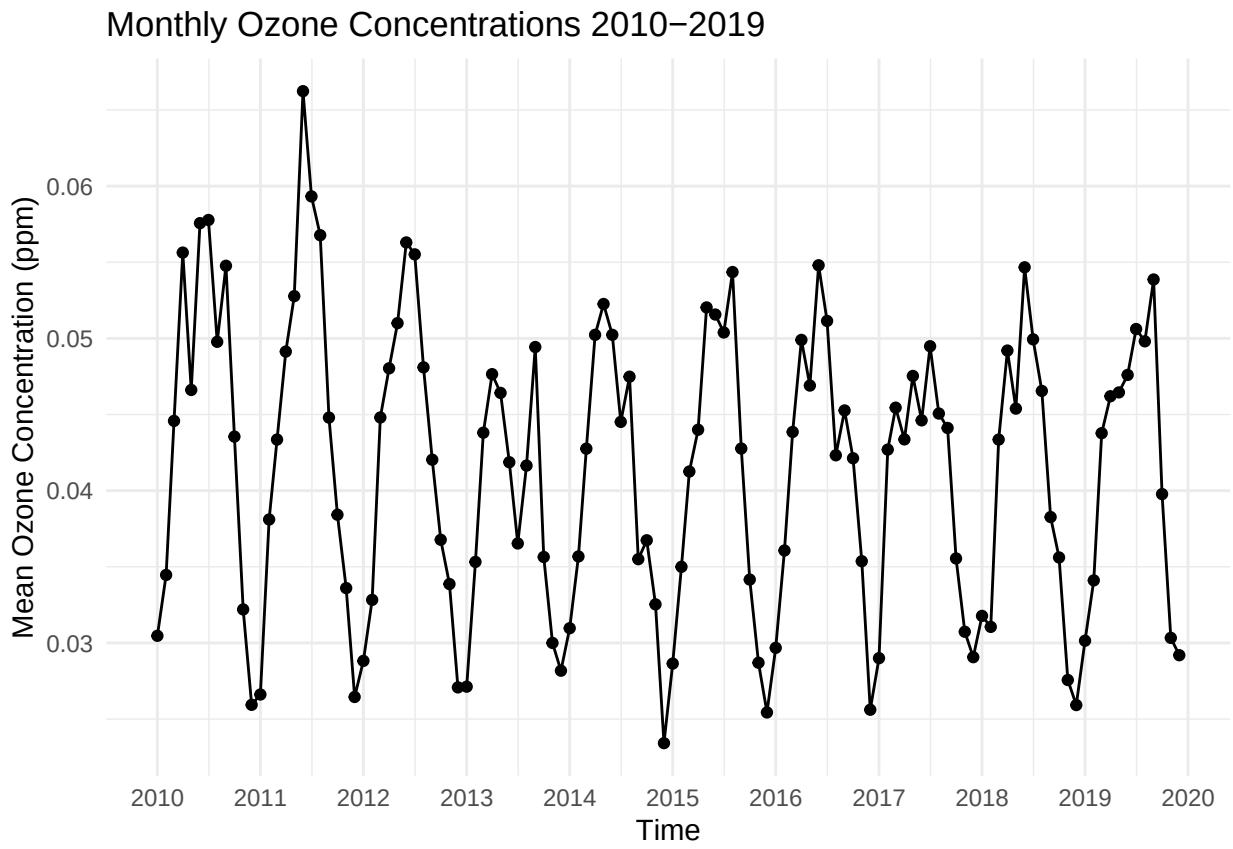
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Seasonal Mann-Kendall test is designed to address seasonal cycles. Using standart test would be insufficient to understand the seasonal variations and trends.

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
#create a point and line graph together of monthly mean ozone by time
Q13 <- ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Monthly Ozone Concentrations 2010-2019",
    x = "Time",
```

```
    y = "Mean Ozone Concentration (ppm)"
    ) +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +  #add years to x scale
  theme_minimal()  #include grid

#print the plot
Q13
```



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: Based on seasonal Mann-Kendall test, there is a slightly decreasing trend in ozone concentrattions where tau=-0.143 and it shows that the result is statistically significant at 95% level. (tau=-0.143 and p-value=0.0467). There is also seasonal variations in the data.

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 the components and turn them into data frames
```

```r
GaringerOzone.monthly_components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

#subtract the seasonal component from the original time series
GaringerOzone.monthly.ts_noseason <- GaringerOzone.monthly.ts - GaringerOzone.monthly_components$seasona

#16
#run the Mann-Kendall test on the deseasonalized time series
Q16 <- MannKendall(GaringerOzone.monthly.ts_noseason)
Q16
```

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

Answer: Both results for seasonal (tau = -0.143, 2-sided pvalue =0.046724) and deseasonalized (tau = -0.165, 2-sided pvalue =0.0075402) there is slight decrease in ozone levels and both are statistically significant. However, when we extract seasonal effects, the declining trend is stronger and it is even statistically significant at 99% level.