Assignment 8: Time Series Analysis

Angelica Rodriguez

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

```
#setting the directory

getwd()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
#Intalling packages and calling libraries

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 1.9.4     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)
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
```

```
#install.packages("trend")
library(stats)
library(trend)
#install.packages("tinytex")
#tinytex::reinstall_tinytex(repository = "illinois")

#Setting the ggplot theme
theme_set(theme_bw())
```

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

file_path <- "/home/guest/EDA_Spring2025/Data/Raw/Ozone_TimeSeries/"

# Definir nombres de archivos
file_names <- c(
  "EPAair_O3_GaringerNC2010_raw.csv",
  "EPAair_O3_GaringerNC2011_raw.csv",
  "EPAair_O3_GaringerNC2012_raw.csv",
  "EPAair_O3_GaringerNC2013_raw.csv",
  "EPAair_O3_GaringerNC2014_raw.csv",
  "EPAair_O3_GaringerNC2015_raw.csv",
  "EPAair_O3_GaringerNC2016_raw.csv",
  "EPAair_O3_GaringerNC2017_raw.csv",
  "EPAair_O3_GaringerNC2018_raw.csv",
  "EPAair_O3_GaringerNC2019_raw.csv"
```

```
)

# Importar y combinar datasets
GaringerOzone <- bind_rows(
  lapply(file_names, function(file) read.csv(paste0(file_path, file), header = TRUE))
)

# Verificar dimensiones del dataframe combinado
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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

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

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

str(GaringerOzone)
```

```
## '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

# To create a sequence of dates from 2010-01-01 to 2019-12-31

start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
```

```r
Days <- as.data.frame(seq(from = start_date, to = end_date, by = "day"))

# Here I create a "Days" dataframe with a complete sequence of dates
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
Days <- as.data.frame(seq(from = start_date, to = end_date, by = "day"))
colnames(Days) <- "Date"

# Fill in missing days
GaringerOzone <- Days %>%
left_join(GaringerOzone, by = "Date")

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

# 6

# Combining the data frames
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")

# Checking the dimensions of the combined data frame

dim(GaringerOzone)
```

```
## [1] 3652    3
```

Visualize

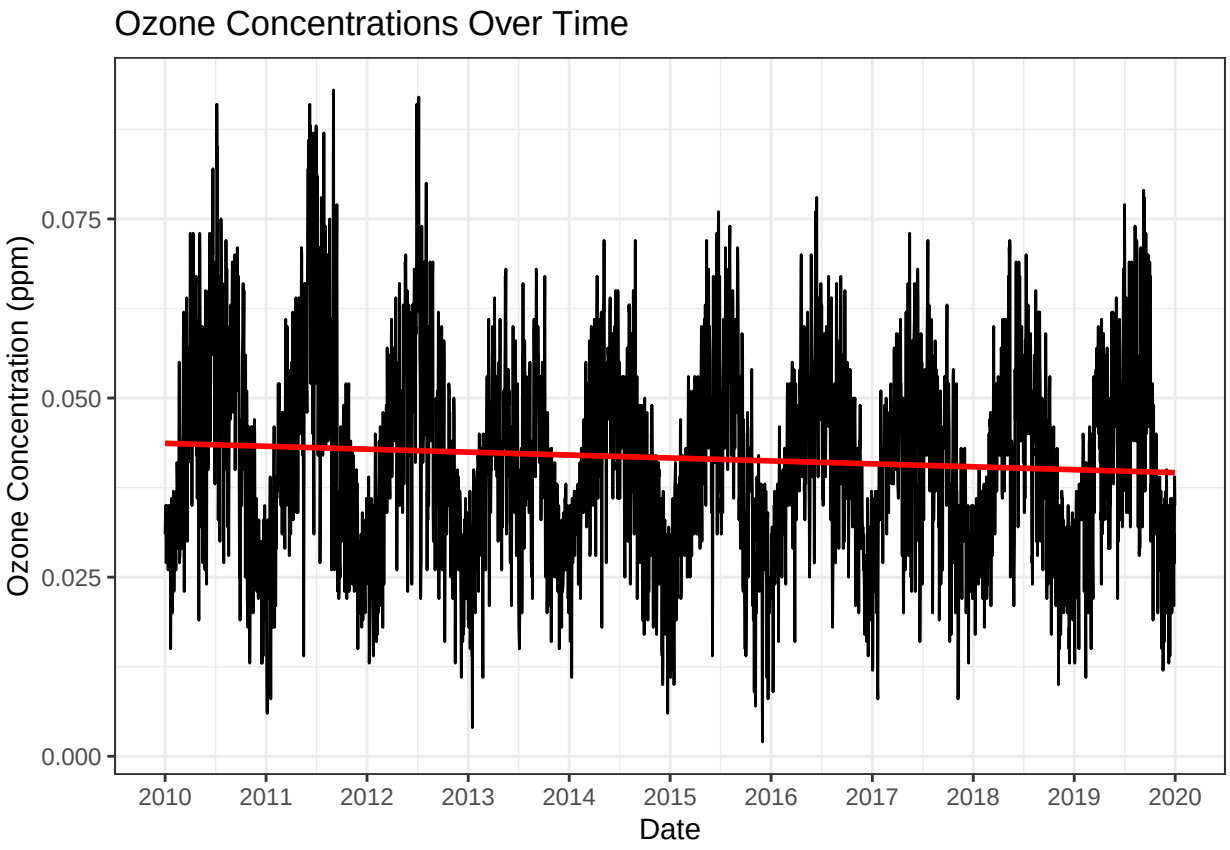
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```r
#7
# Filtering out rows with missing ozone concentration values (NA)
filtered_data <- GaringerOzone %>%
filter(!is.na(Daily.Max.8.hour.Ozone.Concentration))

# Checking if there are any valid data points left
if (nrow(filtered_data) > 0) {

# Creating a line plot with ggplot2
ggplot(filtered_data, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
geom_line() +
geom_smooth(method = "lm", se = FALSE, color = "red") +
labs(
title = "Ozone Concentrations Over Time",
x = "Date",
y = "Ozone Concentration (ppm)"
) +
scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
theme_bw()
} else { cat("No valid data points for the plot.") }
```

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



Answer:This plot shows a slightly downward trend from 2010 to 2020. We can observe seasonal changes and fluctuations for each year, with a peak in ozone concentrations in the middle of the year and a decline at the beginning. Additionally, we can identify where the highest peaks occurred from 2010 to 2013 and the reduction of these peaks from 2014 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

GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, na.rm = FALSE)
```

Answer:This method estimates missing values by fitting a straight line between known data points. The trend line demonstrates a linear pattern and closely aligns with the actual data trend. In contrast, splines typically fit a smooth curve through the data and can introduce unrealistic oscillations, creating trends that diverge from reality in the missing data. Moreover,

the piecewise method prolongs the last observed value of the missing data, assuming that ozone levels remain constant until the next data point, which may not accurately reflect the realities of environmental time series.

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

#Creating the two columns of year and month

GaringerOzone <- GaringerOzone %>%
mutate(year = year(Date),
month = month(Date))


# Adding the data mean ozone concentrations for each month
GaringerOzone.monthly <- GaringerOzone %>%
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.
```

```
# Create a new Date column with each month-year combination set as the first day of the month
GaringerOzone.monthly <- GaringerOzone.monthly %>%
mutate(Date = as.Date(paste(year, month, "01", sep = "-")))

# View the first few rows of the aggregated data
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 4
## # Groups:   year [1]
##    year month mean_ozone Date
##   <dbl> <dbl>      <dbl> <date>
## 1  2010     1     0.0305 2010-01-01
## 2  2010     2     0.0345 2010-02-01
## 3  2010     3     0.0446 2010-03-01
## 4  2010     4     0.0556 2010-04-01
## 5  2010     5     0.0466 2010-05-01
## 6  2010     6     0.0576 2010-06-01
```

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
# Daily Time Series
```

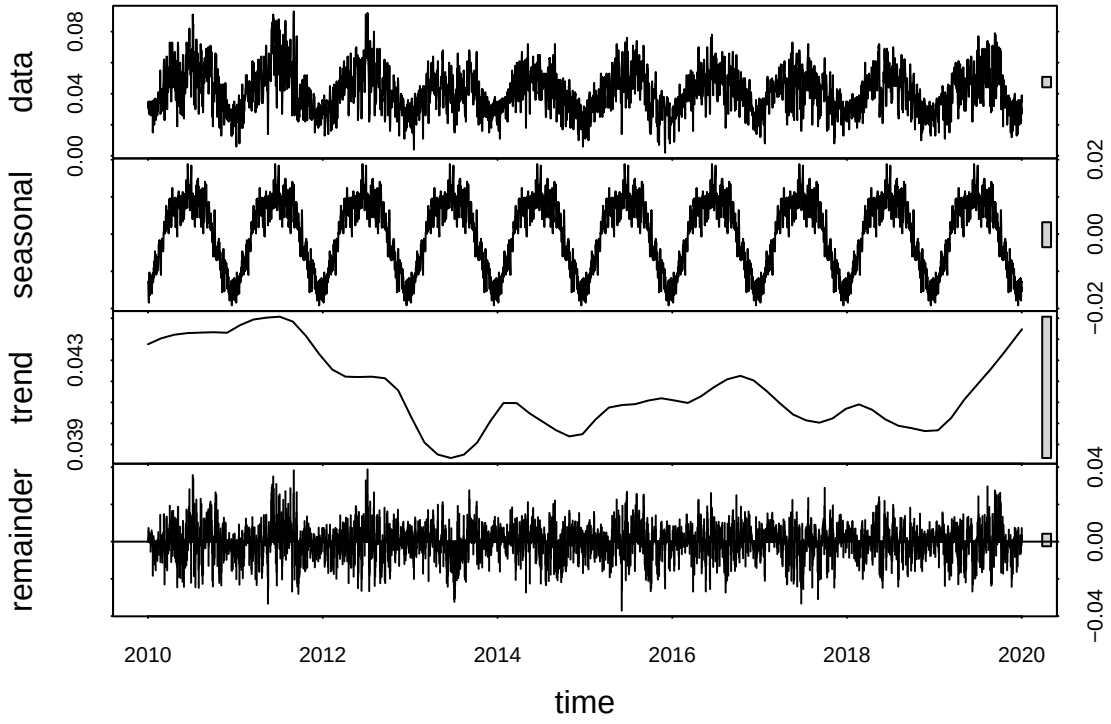
```
start_year_daily <- year(first(GaringerOzone$Date))
monthfirst_daily <- month(first(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
start = c(start_year_daily, monthfirst_daily, 1),
frequency = 365)

# Monthly Time Series
start_year_monthly <- year(first(GaringerOzone.monthly$Date))
monthfirst_monthly <- month(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
start = c(2010, 1),
frequency = 12)
```

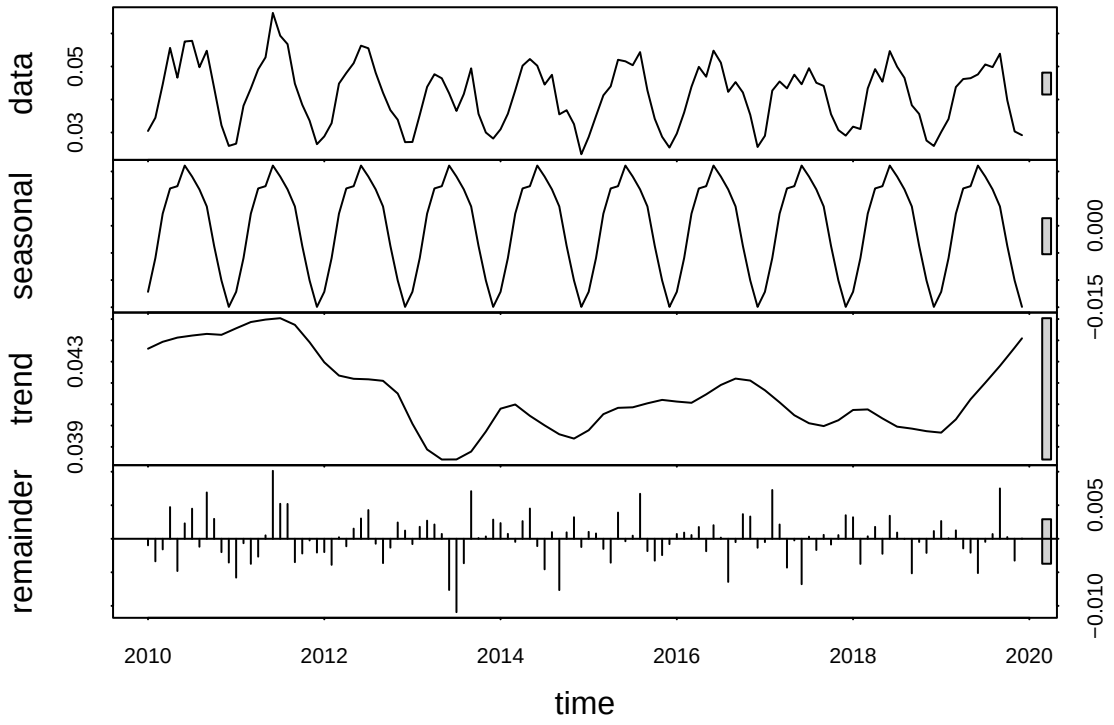
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

#DAILY time series
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
# Plotting the components
plot(GaringerOzone.daily.decomp)
```



```r
#MONTHLY time series
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
# Plotting the components
plot(GaringerOzone.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?

```r
#12

# Running the Seasonal Mann-Kendall test
result <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
print(result)
```

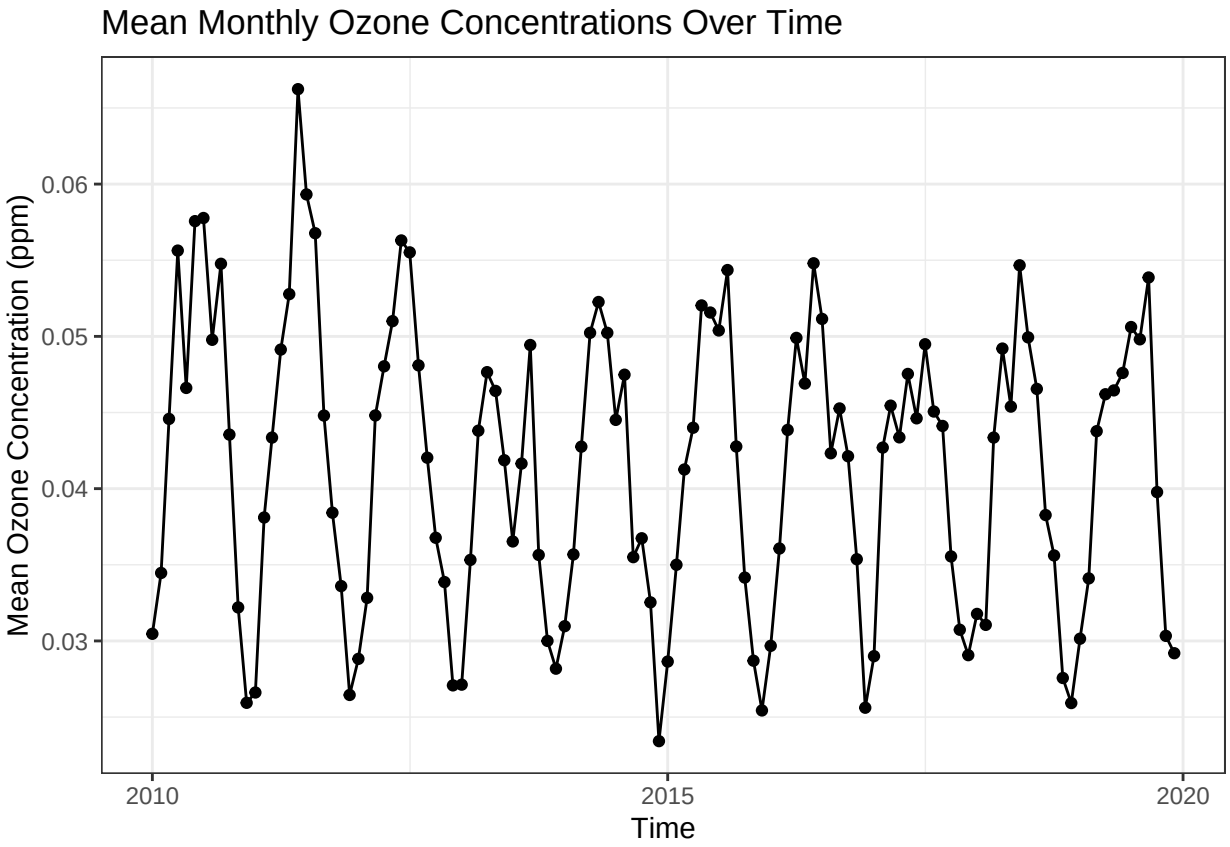
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Ozone levels vary monthly due to meteorological and environmental factors, exhibiting a clear seasonal trend and monthly fluctuations. In this case, the Seasonal Mann-Kendall (SMK) test is applied because it accounts for these seasonal cycles, which a standard Mann-Kendall test might otherwise obscure or misinterpret. The reoults shows a Tau = -0.143 indicating a downward trend in the data. Since this value is close to 0, the trend is weak, although statistically significant at the 5% significance level, showing sufficient evidence to reject the null hypothesis (H0: no trend) and conclude that there is a downward trend in the 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.

```
# 13

plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
geom_point() +
geom_line() +
labs(
title = "Mean Monthly Ozone Concentrations Over Time",
x = "Time",
y = "Mean Ozone Concentration (ppm)"
) +
theme_bw()
print(plot)
```



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: According to the analysis of monthly average Kendal on the deseasonalized data data from July 2010 to February 2019, using the Mann-Kendall Seasonal Trend Test, an oscillating trend in ozone concentration was observed, with a slight but statistically significant decrease at the 5% significance level (tau= -0.143, p-value = 0.046724). This result suggests a possible change in ozone concentration; however, since the observed variability could be due to random

factors, further analysis with additional information and other data is required to determine if specific factors are influencing this 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

# Extracting seasonal component from the decomposition results
GaringerOzone.monthly.components <-
as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

# Subtracting the seasonal component from the original monthly time series
GaringerOzone.monthly.ts_deseasonalized <-
GaringerOzone.monthly.ts - GaringerOzone.monthly.components$seasonal

# Running the Mann-Kendall test on the deseasonalized time series
result_deseasonalized <- MannKendall(GaringerOzone.monthly.ts_deseasonalized)
print(result_deseasonalized)
```

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

```
#16

# Running the Mann-Kendall test on the deseasonalized data
mk_result <- Kendall::MannKendall(GaringerOzone.monthly.ts_deseasonalized)

print(mk_result)
```

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

Answer:Comparing both results with Kendall seasoned (tau = -0.143, 2-side pvalue =0.046724) and Kendall on the deseasonalized data (tau = -0.165, 2-side pvalue =0.0075402) shows that there is a larger decreasing trend in the deseasonalized data. Because Kendall on the deseasonalized data ignores the changes that occur due to the oscillations of the seasons, the decreasing trend in ozone concentration is larger, thus presenting a tau = -0.165 in the deseasonalized data compared to a tau = -0.143 with Kendall seasoned.