Assignment 8: Time Series Analysis

Kat Horan

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A08_TimeSeries.pdf") prior to submission.

The completed exercise is due on Tuesday, 19 March, 2019 before class begins.

Brainstorm a project topic

1. Spend 15 minutes brainstorming ideas for a project topic, and look for a dataset if you are choosing your own rather than using a class dataset. Remember your topic choices are due by the end of March, and you should post your choice ASAP to the forum on Sakai.

Question: Did you do this?

> ANSWER: I spent a little over 15 minutes looking at the data and coming up with a few ideas, but after looking at the forum it seems like other people also had those ideas. I really liked the idea of analyzing the effects of toxins on mortality in the ECOTOX Neonicotinoids Mortality data set, so I am going to look for other data along those lines. I will have this figured out by the time our project ideas are due!

Set up your session

2. Set up your session. Upload the EPA air quality raw dataset for PM2.5 in 2018, and the processed NTL-LTER dataset for nutrients in Peter and Paul lakes. Build a ggplot theme and set it as your default theme. Make sure date variables are set to a date format.

```
# setwd("/Users/kathleenhoran/Desktop/Duke/Spring 2019/Env. Data Analytics/Env_Data_Analytics")

# Load packages:
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(nlme)
library(lsmeans)
```

```
## Loading required package: emmeans
```

```
## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.
```

```
library(multcompView)
library(trend)
library(ggplot2)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  2.0.1       v purrr   0.3.0
## v tidyr   0.8.2       v dplyr   0.8.0.1
## v readr   1.3.1       v stringr 1.4.0
## v tibble  2.0.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x dplyr::collapse()        masks nlme::collapse()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
# Load EPA air quality data for PM2.5 in 2018:
EPAair_PM25_NC18_raw <-
  read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv")

## Convert date:
EPAair_PM25_NC18_raw$Date <- as.Date(EPAair_PM25_NC18_raw$Date, format = "%m/%d/%y")

## Confrim date is converted:
class(EPAair_PM25_NC18_raw$Date)
```

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

```
# Load processed nutrient data for Peter & Paul lakes:
PeterPaul.nutrients.processed <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed

## Convert date:
PeterPaul.nutrients.processed$sampledate <- as.Date(PeterPaul.nutrients.processed$sampledate, format =
```

```
## Confrim date is converted:
class(PeterPaul.nutrients.processed$sampledate)
```

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

```
# Set theme:
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

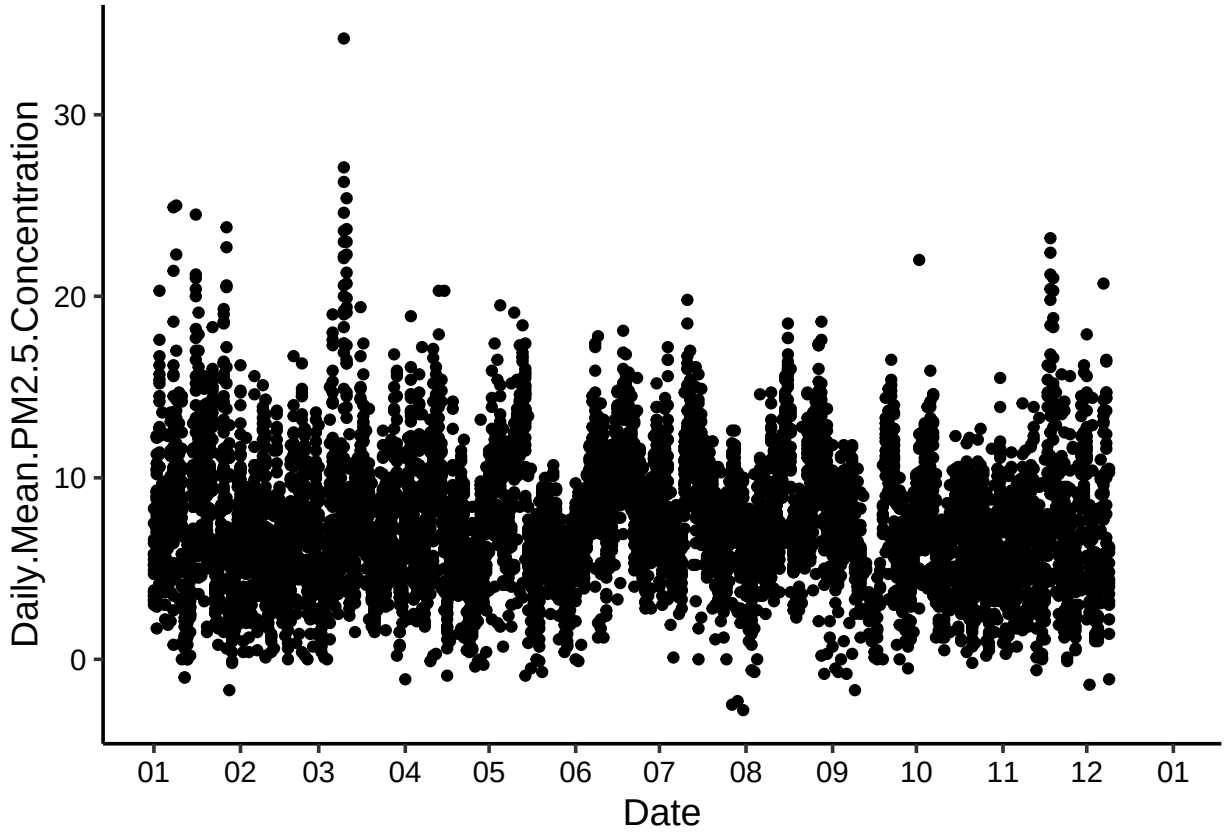
Run a hierarchical (mixed-effects) model

Research question: Do PM2.5 concentrations have a significant trend in 2018?

3. Run a repeated measures ANOVA, with PM2.5 concentrations as the response, Date as a fixed effect, and Site.Name as a random effect. This will allow us to extrapolate PM2.5 concentrations across North Carolina.

3a. Illustrate PM2.5 concentrations by date. Do not split aesthetics by site.

```
ggplot(EPAair_PM25_NC18_raw, aes(x = Date, y = Daily.Mean.PM2.5.Concentration)) +
  scale_x_date(limits = as.Date(c("2018-01-01", "2018-12-31")),
          date_breaks = "1 month", date_labels = "%m") +
  geom_point()
```



3b. Insert the following line of code into your R chunk. This will eliminate duplicate measurements on single dates for each site.

PM2.5 = PM2.5[order(PM2.5[,'Date'],-PM2.5[,'Site.ID']),] PM2.5 = PM2.5[!duplicated(PM2.5$Date),]

3c. Determine the temporal autocorrelation in your model.

3d. Run a mixed effects model.

```
PM2.5 <- EPAair_PM25_NC18_raw
PM2.5 = PM2.5[order(PM2.5[,'Date'],-PM2.5[,'Site.ID']),]
PM2.5 = PM2.5[!duplicated(PM2.5$Date),]

# Auto test
PM2.5.Test.auto <- lme(data = PM2.5,
                       Daily.Mean.PM2.5.Concentration ~ Date, random = ~1|Site.ID)
PM2.5.Test.auto
```

```
## Linear mixed-effects model fit by REML
##   Data: PM2.5
##   Log-restricted-likelihood: -928.6076
##   Fixed: Daily.Mean.PM2.5.Concentration ~ Date
##  (Intercept)         Date
## 90.465022634 -0.004727976
##
## Random effects:
##  Formula: ~1 | Site.ID
##         (Intercept) Residual
## StdDev:    1.650184 3.559209
##
## Number of Observations: 343
## Number of Groups: 3
```

```
ACF(PM2.5.Test.auto) # ACF of .514
```

```
##    lag          ACF
## 1    0  1.000000000
## 2    1  0.513829909
## 3    2  0.194512680
## 4    3  0.117925187
## 5    4  0.126462863
## 6    5  0.100699787
## 7    6  0.058215891
## 8    7 -0.053090104
## 9    8  0.017671857
## 10   9  0.012177847
## 11  10 -0.003699721
## 12  11 -0.020305291
## 13  12 -0.044621086
## 14  13 -0.055602646
## 15  14 -0.065787345
## 16  15 -0.123987593
## 17  16 -0.055414056
## 18  17  0.002911218
## 19  18  0.025133456
## 20  19 -0.015306468
## 21  20 -0.143472007
## 22  21 -0.155495492
## 23  22 -0.060369985
## 24  23  0.003954231
```

```
## 25  24  0.042295682
## 26  25  0.001320007
```

```
PM2.5.Test.mixed <- lme(data = PM2.5,
                    Daily.Mean.PM2.5.Concentration ~ Date, random = ~1|Site.ID,
                    correlation = corAR1(form = ~ Date|Site.ID, value = 0.514),
                    method = "REML")
summary(PM2.5.Test.mixed)
```

```
## Linear mixed-effects model fit by REML
##  Data: PM2.5
##        AIC      BIC   logLik
##   1756.622 1775.781 -873.311
##
## Random effects:
##  Formula: ~1 | Site.ID
##         (Intercept) Residual
## StdDev: 0.001028133 3.597269
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~Date | Site.ID
##  Parameter estimate(s):
##      Phi1
## 0.5384349
## Fixed effects: Daily.Mean.PM2.5.Concentration ~ Date
##                Value Std.Error  DF   t-value p-value
## (Intercept) 83.14801  60.63585 339  1.371268  0.1712
## Date        -0.00426   0.00342 339 -1.244145  0.2143
##  Correlation:
##      (Intr)
## Date -1
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -2.3220745 -0.6187194 -0.1116751  0.6164257  3.4192603
##
## Number of Observations: 343
## Number of Groups: 3
```

Is there a significant increasing or decreasing trend in PM2.5 concentrations in 2018?

> ANSWER: There is a decreasing trend in PM2.5 concentrations. For each increase in date, the units of PM2.5 concentration decrease by -0.00426.

3e. Run a fixed effects model with Date as the only explanatory variable. Then test whether the mixed effects model is a better fit than the fixed effect model.

```
PM2.5.Test.fixed <- gls(data = PM2.5,
                      Daily.Mean.PM2.5.Concentration ~ Date,
                     method = "REML")
summary(PM2.5.Test.fixed)
```

```
## Generalized least squares fit by REML
##   Model: Daily.Mean.PM2.5.Concentration ~ Date
##   Data: PM2.5
##        AIC      BIC    logLik
##   1865.202 1876.698 -929.6011
```

```
##
## Coefficients:
##                Value Std.Error   t-value p-value
## (Intercept) 98.57796  34.60285  2.848840  0.0047
## Date        -0.00513   0.00195 -2.624999  0.0091
##
##  Correlation:
##      (Intr)
## Date -1
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.3531000 -0.6348100 -0.1153454  0.6383004  3.4063068
##
## Residual standard error: 3.584321
## Degrees of freedom: 343 total; 341 residual
```

```
anova(PM2.5.Test.mixed, PM2.5.Test.fixed)
```

```
##                   Model df      AIC      BIC    logLik   Test  L.Ratio
## PM2.5.Test.mixed      1  5 1756.622 1775.781 -873.3110
## PM2.5.Test.fixed      2  3 1865.202 1876.698 -929.6011 1 vs 2 112.5802
##                   p-value
## PM2.5.Test.mixed
## PM2.5.Test.fixed  <.0001
```

Which model is better?

> ANSWER: The mixed effect is better as it has the lower AIC of 1756.6 versus the AIC of 1865.2 of the fixed effect model. The small p-value of the fixed effect model also shows us that the models have a significantly different fit.

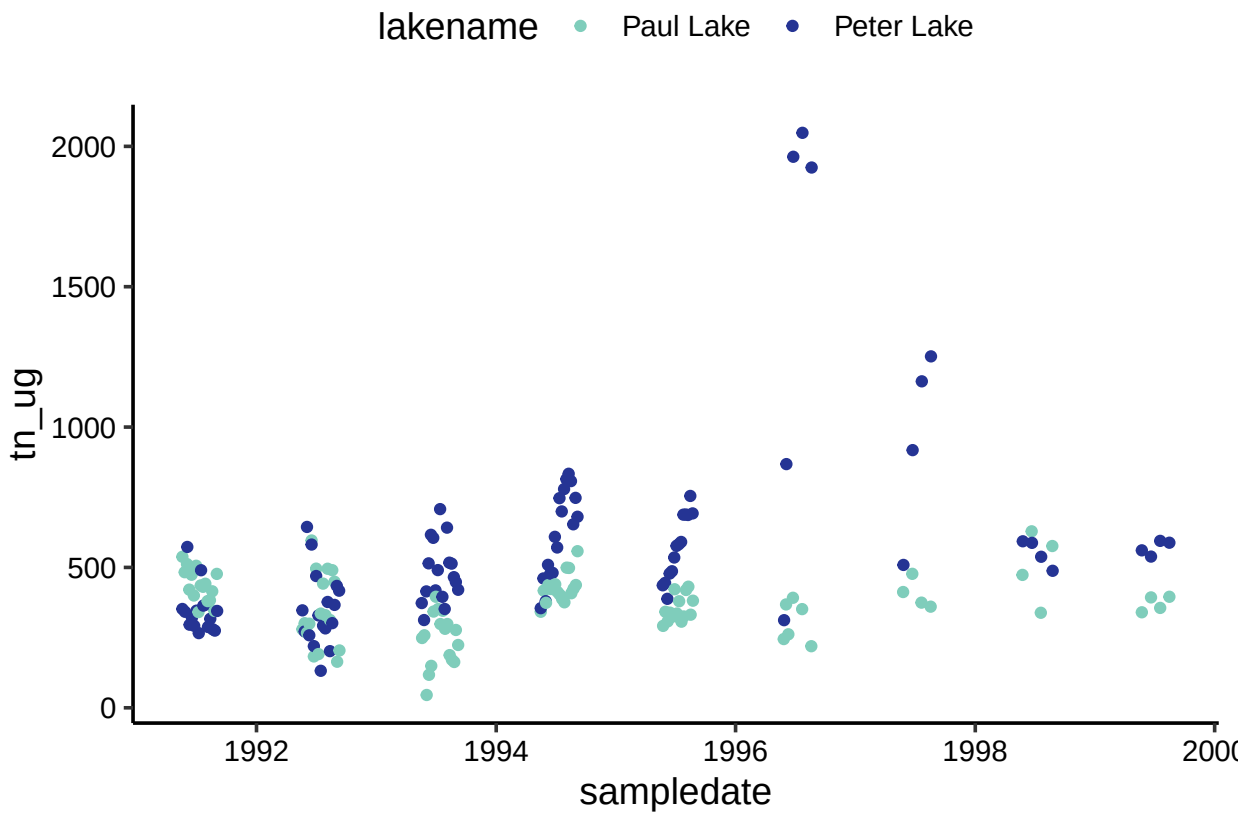
Run a Mann-Kendall test

Research question: Is there a trend in total N surface concentrations in Peter and Paul lakes?

4. Duplicate the Mann-Kendall test we ran for total P in class, this time with total N for both lakes. Make sure to run a test for changepoints in the datasets (and run a second one if a second change point is likely).

```
# Wrangle data:
PeterPaul.nutrients.processed <-
  PeterPaul.nutrients.processed %>%
  select(-lakeid, -depth_id, -comments) %>%
  filter(depth == 0) %>%
  filter(!is.na(tn_ug))

# Initial visualization of data
ggplot(PeterPaul.nutrients.processed, aes(x = sampledate, y = tn_ug, color = lakename)) +
  geom_point() +
  scale_color_manual(values = c("#7fcdbb", "#253494"))
```




```
# Split data by lake:
Peter.nutrients.processed <- filter(PeterPaul.nutrients.processed, lakename == "Peter Lake")
Paul.nutrients.processed <- filter(PeterPaul.nutrients.processed, lakename == "Paul Lake")

# Check Peter Lake data
# Run MK test:
mk.test(Peter.nutrients.processed$tn_ug)
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.processed$tn_ug
## z = 7.2927, n = 98, p-value = 3.039e-13
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 2.377000e+03 1.061503e+05 5.001052e-01
```

```
# Test for change point:
pettitt.test(Peter.nutrients.processed$tn_ug) #says 36
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.processed$tn_ug
## U* = 1884, p-value = 3.744e-10
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
```

```
##                              36
```

```
# Run separate Mann-Kendall for each change point
mk.test(Peter.nutrients.processed$tn_ug[1:35]) #z score is negative
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.processed$tn_ug[1:35]
## z = -0.22722, n = 35, p-value = 0.8203
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
##  -17.00000000 4958.33333333   -0.02857143
```

```
mk.test(Peter.nutrients.processed$tn_ug[36:98]) #z score is positive
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.processed$tn_ug[36:98]
## z = 3.1909, n = 63, p-value = 0.001418
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 5.390000e+02 2.842700e+04 2.759857e-01
```

```
# Is there a second change point?
pettitt.test(Peter.nutrients.processed$tn_ug[36:98])
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.processed$tn_ug[36:98]
## U* = 560, p-value = 0.001213
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              21
```

```
# 36 + 21 = 57

# Run another Mann-Kendall for the second change point
mk.test(Peter.nutrients.processed$tn_ug[36:56]) # z score negative
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.processed$tn_ug[36:56]
## z = -1.0569, n = 21, p-value = 0.2906
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
##  -36.0000000 1096.6666667   -0.1714286
```

```
mk.test(Peter.nutrients.processed$tn_ug[57:98]) # z score positive
```

```
##
```

```
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.processed$tn_ug[57:98]
## z = 0.15172, n = 42, p-value = 0.8794
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
##   15.0000000 8514.3333333    0.0174216
```

```
# Are there additional change points?
pettitt.test(Peter.nutrients.processed$tn_ug[57:98])
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.processed$tn_ug[57:98]
## U* = 127, p-value = 0.5584
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              11
```

```
# Yes, however p-value was high so not significant.

# Check Paul Lake data
# Run MK test:
mk.test(Paul.nutrients.processed$tn_ug) # z score negative
```

```
##
##  Mann-Kendall trend test
##
## data:  Paul.nutrients.processed$tn_ug
## z = -0.35068, n = 99, p-value = 0.7258
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.170000e+02  1.094170e+05 -2.411874e-02
```

```
# Test for change point:
pettitt.test(Paul.nutrients.processed$tn_ug)
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Paul.nutrients.processed$tn_ug
## U* = 704, p-value = 0.09624
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              16
```

```
# Yes, however p-value is high so not significant.
```

What are the results of this test?

> ANSWER: These tests show us that there are significant change points in the data between June 2, 1993 and June 29, 1994 for Peter Lake. Between these two dates there is a negative change. Following June of 1994 there is a positive change.

5. Generate a graph that illustrates the TN concentrations over time, coloring by lake and adding vertical line(s) representing changepoint(s).

```
ggplot(PeterPaul.nutrients.processed, aes(x = sampledate, y = tn_ug, color = lakename)) +
  geom_vline(xintercept = as.Date("1993-06-02"), color = "#253494", lty = 2) +
  geom_vline(xintercept = as.Date("1994-06-29"), color = "#253494", lty = 2) +
  geom_point() +
  scale_color_manual(values = c("#7fcdbb", "#253494"))
```