Assignment 8: Time Series Analysis

Keith Bollt

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: Yes (looking at thermocline change over time)

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.

```
getwd()
```

```
## [1] "V:/ENV_872_Project_Directory/Assignments"
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2
```

```
## -- Attaching packages ----------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
## Warning: package 'readr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.2
## -- Conflicts ------------------------------------------------------------------------------------ ti
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(lubridate)
## Warning: package 'lubridate' was built under R version 3.5.2
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
library(nlme)
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
library(lsmeans)
## Warning: package 'lsmeans' was built under R version 3.5.2
## Loading required package: emmeans
## Warning: package 'emmeans' was built under R version 3.5.2
## 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)
## Warning: package 'multcompView' was built under R version 3.5.2
#install.packages("trend")
library(trend)
## Warning: package 'trend' was built under R version 3.5.2
EPA.air.raw <- read.csv("V:/ENV_872_Project_Directory/Data/Raw/EPAair_PM25_NC2018_raw.csv")
PeterPaul.processed <- read.csv("V:/ENV_872_Project_Directory/Data/Processed/NTL-LTER_Lake_Chemistry_Nu
View(EPA.air.raw)
View(PeterPaul.processed)

mytheme <- theme_classic(base_size = 13) +
```

```
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)

PeterPaul.processed$sampledate <- as.Date(PeterPaul.processed$sampledate,format = "%Y-%m-%d")
EPA.air.raw$Date <- as.Date(EPA.air.raw$Date, format = "%m/%d/%y")
```

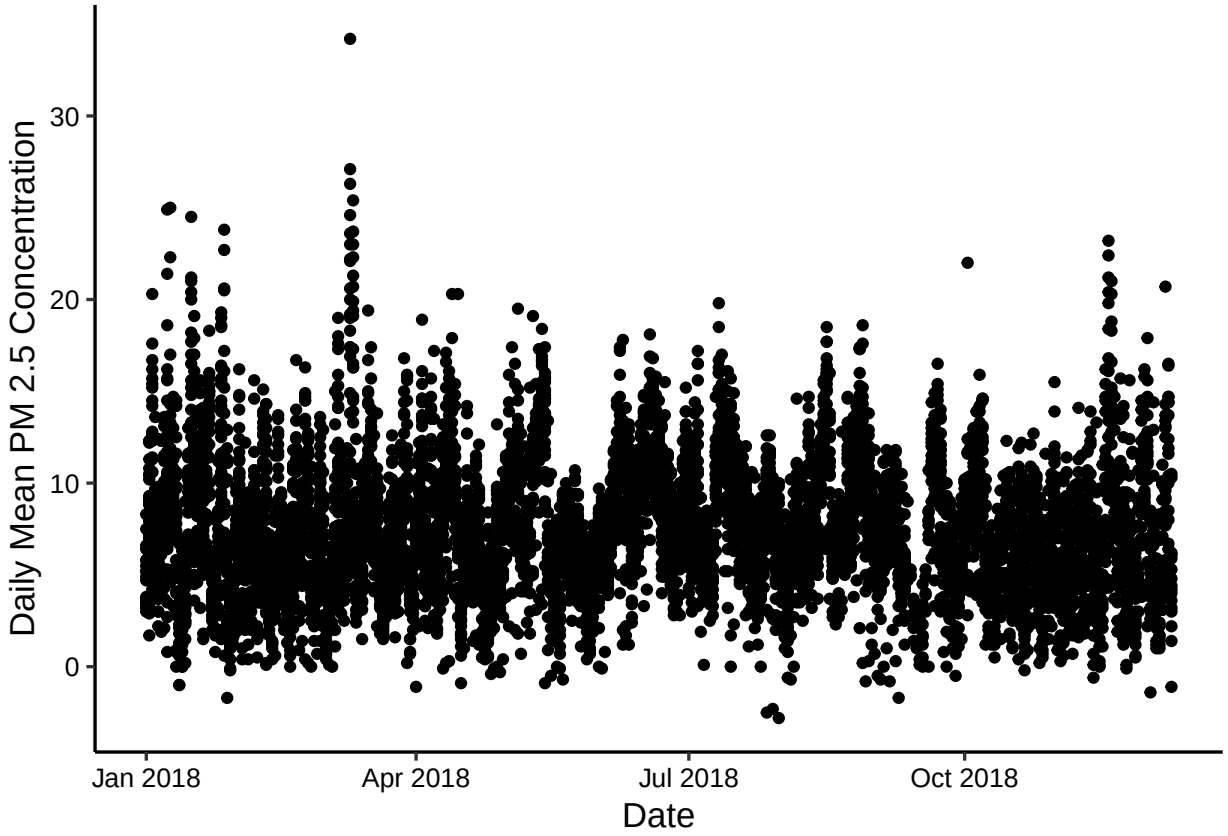
Run a hierarchical (mixed-effects) model

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.

```
PM2.5_plot <- ggplot(EPA.air.raw, aes(x = Date, y = Daily.Mean.PM2.5.Concentration))+
  geom_point()+
  labs(x= "Date", y = "Daily Mean PM 2.5 Concentration")
PM2.5_plot
```



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.

```
EPA.air.raw = EPA.air.raw[order(EPA.air.raw[,'Date'],-EPA.air.raw[,'Site.ID']),]
EPA.air.raw = EPA.air.raw[!duplicated(EPA.air.raw$Date),]

TempTest.HW.auto <- lme(data = EPA.air.raw,
                     Daily.Mean.PM2.5.Concentration ~ Date,
                     random = ~1|Site.Name)
TempTest.HW.auto
```

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

```
ACF(TempTest.HW.auto)
```

```
##    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
```

```
# 51.38% autocorrelation

TempTest.HW.mixed <- lme(data = EPA.air.raw,
                     Daily.Mean.PM2.5.Concentration ~ Date,
                     random = ~1|Site.Name,
                     correlation = corAR1(form = ~ Date|Site.Name, value = 0.5138), #correlation from p
                     #define method as restricted maximum likelihood
                     method = "REML")
summary(TempTest.HW.mixed)
```

```
## Linear mixed-effects model fit by REML
##  Data: EPA.air.raw
##        AIC      BIC   logLik
##   1756.622 1775.781 -873.311
##
## Random effects:
##  Formula: ~1 | Site.Name
##         (Intercept) Residual
## StdDev:  0.00103013 3.597269
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~Date | Site.Name
##  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: No, there is not. The two variables tested generated p-values of 0.17 and 0.21, respectively. This means we cannot reject the null hypothesis that there is not a significant trend in PM2.5 concentrations in 2018.

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.

```
TempTest.HW.fixed <- gls(data = EPA.air.raw,
                     Daily.Mean.PM2.5.Concentration ~ Date,
                     method = "REML")
anova(TempTest.HW.mixed, TempTest.HW.fixed)
```

```
##                   Model df      AIC      BIC    logLik   Test  L.Ratio
## TempTest.HW.mixed     1  5 1756.622 1775.781 -873.3110
```

```
## TempTest.HW.fixed     2  3 1865.202 1876.698 -929.6011 1 vs 2 112.5802
##                    p-value
## TempTest.HW.mixed
## TempTest.HW.fixed   <.0001
```

Which model is better?

> ANSWER: The mixed model is better. It has an AIC of 1756, which is quite a bit lower than the fixed model's AIC of 1865. In addition, the p-value for whether there is a significantly different fit between the two models is <0.05, which means that we can reject the null hypothesis that they do not 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?

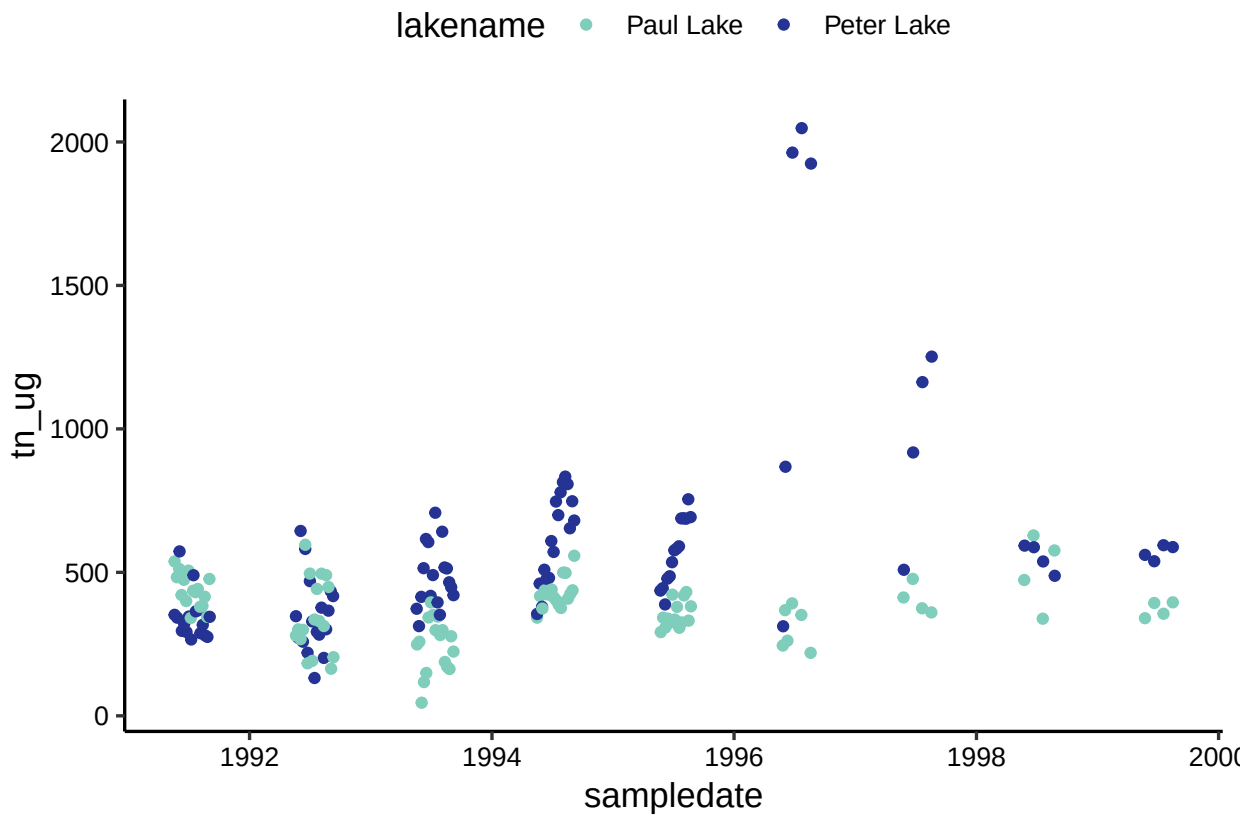
4. Duplicate the Mann-Kendall test we ran for total P in class, this time with total N for both lakes. Make sure to run a test for changepoints in the datasets (and run a second one if a second change point is likely).

```
PeterPaul.nutrients.surface <-
  PeterPaul.processed %>%
  filter(depth == 0) %>%
  filter(!is.na(tn_ug))
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

```
Peter.nutrients.surface <- filter(PeterPaul.nutrients.surface, lakename == "Peter Lake")
Paul.nutrients.surface <- filter(PeterPaul.nutrients.surface, lakename == "Paul Lake")

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




```
#Peter Lake
mk.test(Peter.nutrients.surface$tn_ug)
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$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
```

```
pettitt.test(Peter.nutrients.surface$tn_ug)  #break at row 36
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.surface$tn_ug
## U* = 1884, p-value = 3.744e-10
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              36
```

```
mk.test(Peter.nutrients.surface$tn_ug[1:35])  # no trend
```

```
##
```

```
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$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.surface$tn_ug[36:98]) #significant trend
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$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
```

```
pettitt.test(Peter.nutrients.surface$tn_ug[36:98]) #break at row 36+21 = 57,
```

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

```
#A break in the data in June 1994 doesn't really make sense, because nothing changed in how the lake wa

#Paul Lake
mk.test(Paul.nutrients.surface$tn_ug) # no trend
```

```
##
##  Mann-Kendall trend test
##
## data:  Paul.nutrients.surface$tn_ug
## z = -0.1572, n = 99, p-value = 0.8751
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -5.300000e+01  1.094170e+05 -1.092558e-02
```

```
pettitt.test(Paul.nutrients.surface$tn_ug) # You can still generate breaks in your data even if no brea
```

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

```
## probable change point at time K
##                              16
```

What are the results of this test?

> ANSWER: The Mann-Kendall test for Peter Lake demonstrates that there is one break in the data that both is borne out of the data and makes sense knowing how Peter Lake was managed in the 1990s. This break took place on June 2, 1993. The Mann-Kendall test for Paul Lake showed there is not a significant trend in the data. Given what I know about how Paul Lake was managed in the 1990s, there is no reason to suspect that the breakline suggested by the Pettitt test for Paul Lake represents an actual break in the data.

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.surface, aes(x = sampledate, y = tn_ug, color = lakename)) +
  geom_point() +
  geom_vline(xintercept= as.Date("1993-06-02"), color = "#253494")+
  scale_color_manual(values = c("#7fcdbb", "#253494"))+
  labs(x = "Date", y= "Nitrogen Concentration", color = "Lake Name")
```