Time Series Introduction: Sesonal Decomposition

Paul Hanson, pchanson@wisc.edu, adapted by: Veronica Gibson

2025-08-01

Time Series Analysis Introduction: Seasonal Decomposition

Project: NSC SWMP Synthesis Catalyst Class

Funding: NERRS Science Collaborative

Author(s): Paul Hanson, pchanson@wisc.edu

University of Wisconsin-Madison, Center for Limnology

Adapted to R markdown by Veronica Gibson, University of Hawaii at Manoa, HIMB

Data originated from NERRS CDMO: https://cdmo.baruch.sc.edu/ and data was compiled into csvs for all Reserves using code from:

https://github.com/Lake-Superior-Reserve/WQ_SWMP_Synthesis/tree/main/R/Data_processing This repository is currently private until analyses are complete, but will be made public after publication.

This code uses data from Apalachicola Reserve. This folder has been uploaded to the repo. Code showing how data was split out for Reserves is in 'Data Processing'. This code selects a single variable from a single reserve site, based on user input, and separates the trend (i.e., long-term variability), seasonal component (i.e., annual cycle) and random component of the signal (variable). Note that the seasonal component is the calculated "average seasonal" signal, so each year in a multi-year data set will have the same seasonal component. All the variation is in the trend and random components. After decomposition, each of the three components are run through the ACF to demonstrate #the degree of autcorrelation in the components.

Data set up

```r
# Load the data into separate data frames
datMet<-read.csv("APA/met_apa.csv")
datNut<-read.csv("APA/nut_apa.csv")
datWQ<-read.csv("APA/wq_apa.csv")

# Create a handy year fraction variable for each data frame for plotting
datMet$YearFrac = datMet$year + datMet$month/12
datNut$YearFrac = datNut$year + datNut$month/12
datWQ$YearFrac  = datWQ$year + datWQ$month/12

# Each reserve can have multiple sampling stations
# Determine the number of stations for each data frame
uMet = unique(datMet$station)
uNut = unique(datNut$station)
uWQ  = unique(datWQ$station)
```

```
# Print out the unique stations for each data frame
cat('Unique met stations: ',uMet,'\n')
```

```
## Unique met stations:  apaebmet
```

```
cat('Unique nut stations: ',uNut,'\n')
```

```
## Unique nut stations:  apacpnut apadbnut apaebnut apaegnut apaesnut apambnut apanhnut apapcnut aparvnu
```

```
cat('Unique WQ  stations: ',uWQ,'\n')
```

```
## Unique WQ  stations:  apabpwq apacpwq apadbwq apaebwq apaeswq apalmwq apapcwq
```

User Input Section

The following code assumes water quality (WQ) data, but can be edited for nutrient or met data.

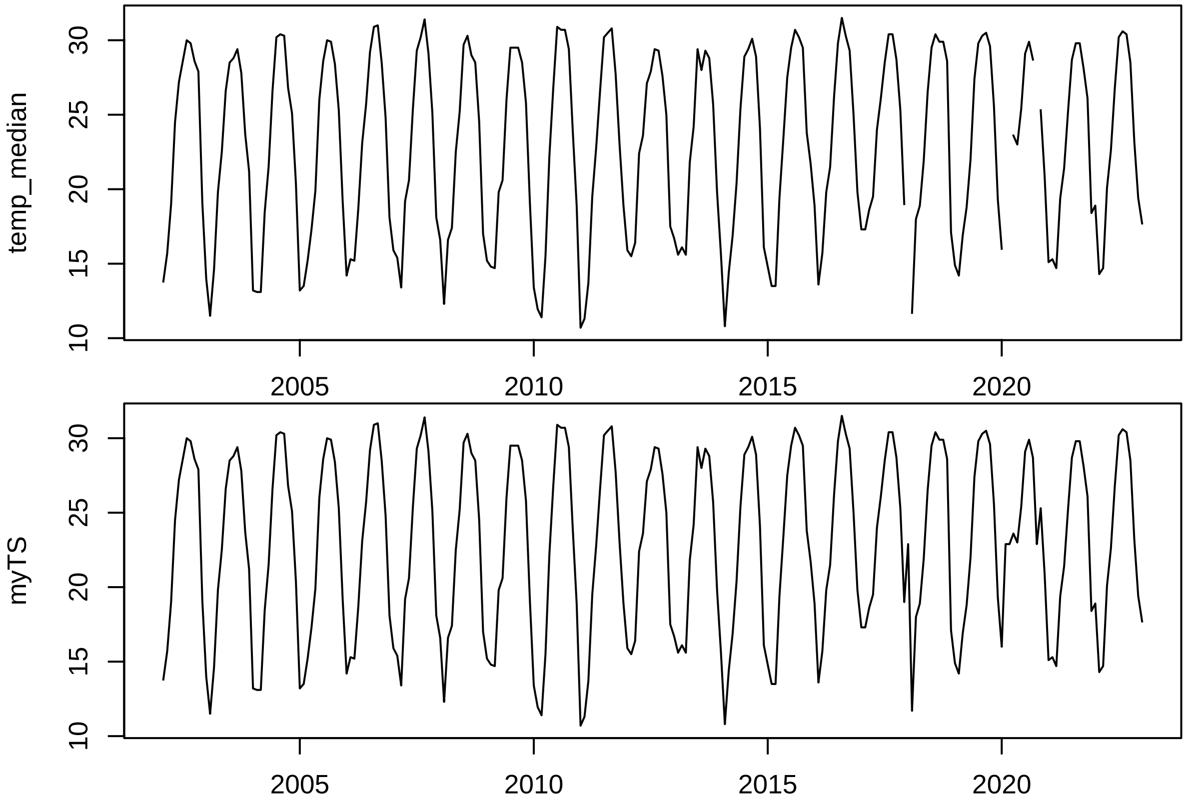
```
#########################################################################################
# Begin user input section

# Load required libraries
library(here)        # Provides an easy way to construct file paths
```

```
## here() starts at /Users/Hulali/Documents/SWMPCourse/2025FallSWMPCourse
```

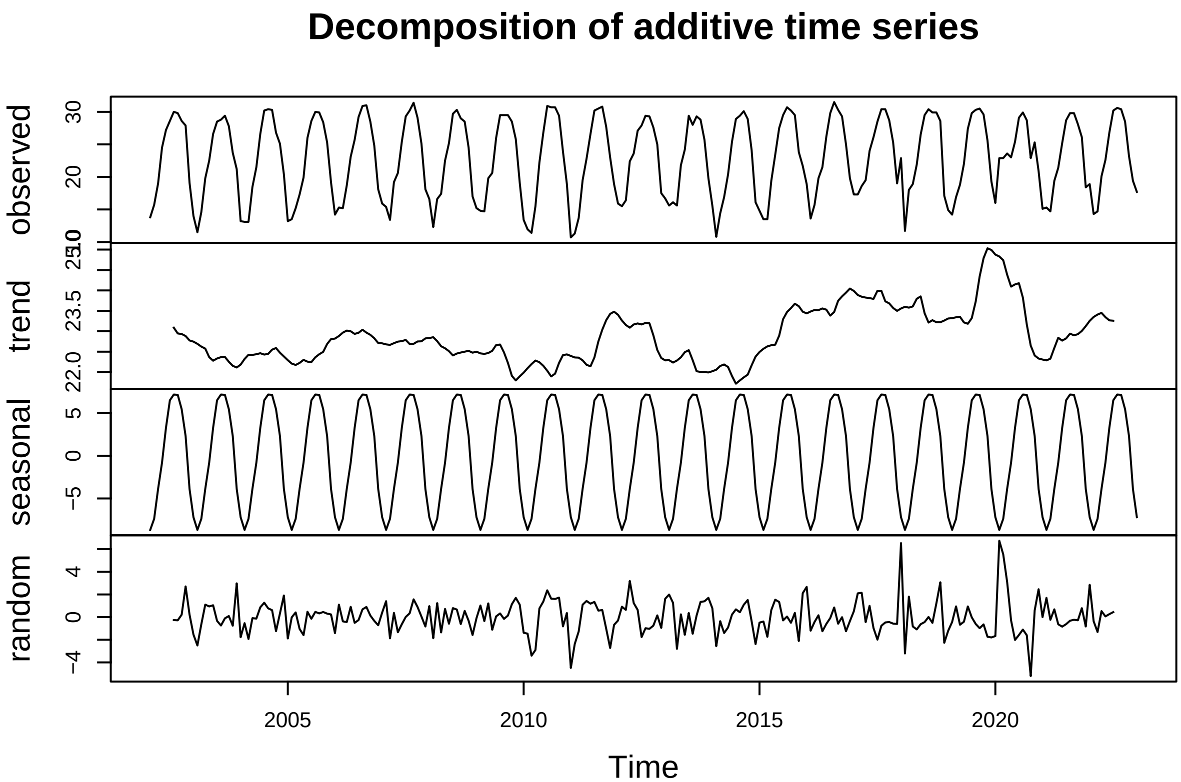
```
# The following two parameters are not necessarily known apriori; code above prints the unique stations
# Select the nth station from the site

#This will be used for plotting the data below

nSta = 2
# Select the nth variable
nCol = 7 # 65 is turb, 37 is DO mean, 43 is depth


whichRows = which(datWQ$station==uWQ[nSta])
# End user input section
###########################
```

Plotting the data

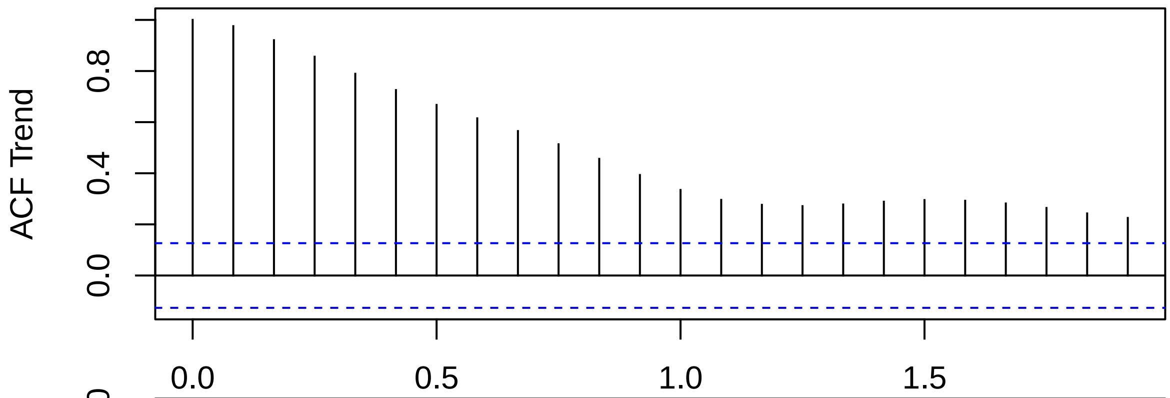
```
# Plot the original data
par(mfrow=c(2,1),lend=2,mai = c(0.25,0.75, 0.08, 0.05),oma = c(2,1,0.2,0.2), cex = 0.8)
myDS = data.frame(YearFrac=datWQ$YearFrac[whichRows],myData = datWQ[whichRows,nCol])
plot(myDS,type='l',xlab = 'Year',ylab=colnames(datWQ[nCol]))

# Create a timeseries object
myTS <- ts(myDS[,2], myDS[1,1], frequency=12)

# find and replace NAs with mean of the time series; other infill techniques could be used
iNA = which(is.na(myTS))
myTS[iNA] = mean(myTS,na.rm=TRUE)

# Plot the time series to compare with the original data
plot(myTS)
```

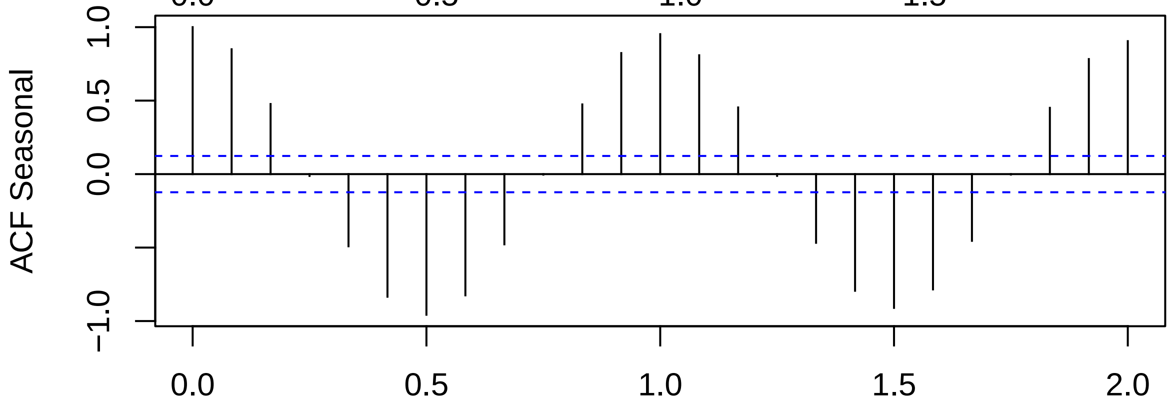



```
# Decompose timeseries into trend, seasonal, and random components
myTSdecomposed = decompose(myTS)
# Plot the decomposed timeseries
plot(decompose(myTS))
```




```r
# Plot the autocorrelation function (ACF) for the trend component of the decomposed time series
# Interpretation: The ACF plot for the trend shows how strongly the current value is correlated
# with past values at various time lags. A slow decay of the ACF suggests a long-term
# dependency in the data.
acf(na.omit(myTSdecomposed$trend), ylab='ACF Trend')

# Plot the autocorrelation function for the seasonal component
# Interpretation: The ACF plot for the seasonal component will often show a periodic pattern,
# indicating that the data repeats at regular intervals (e.g., annual cycles). Peaks at regular
#lags
# suggest strong seasonality.
acf(na.omit(myTSdecomposed$seasonal), ylab='ACF Seasonal')
```






```
# Plot the autocorrelation function for the random (residual) component
# Interpretation: The ACF plot for the random component should ideally show no significant
# correlation at any lag, as the residuals are expected to be random noise. Any significant
# correlations
# might suggest some underlying pattern or structure still remaining in the residuals.
acf(na.omit(myTSdecomposed$random), ylab='ACF Random')
```