Lab 5: Outliers and Missing Data

Luana Lima

02/17/2022

## Objectives

1. Answer questions on M4/M5
2. Show additional ways to handle outliers and missing data
3. Work with another dataset

## Setting R code chunk options

First R code chunk is used for setting the options for all R code chunks. The choice echo=TRUE means both code and output will appear on report, include = FALSE neither code nor output is printed.

## Loading packages and initializing

Second R code chunk is for loading packages. By setting message = FALSE, the code will appear but not the output.

library(lubridate)  
library(ggplot2)  
library(forecast)  
library(Kendall)  
library(tseries)  
  
#New packages for M5  
#install.packages("outliers")  
library(outliers)  
#install.packages("tidyverse")  
library(tidyverse)

## Importing and preparing data

Today we will work with wind speed data.

#Importing time series data from text file#  
wind\_data <- read.csv(file="../Data/Wind\_Speed\_PortArthurTX\_Processed.csv",header=TRUE,stringsAsFactors = TRUE)  
  
#creating data object  
wind\_data$DATE <- ym(wind\_data$DATE)

## Transforming data into time series object

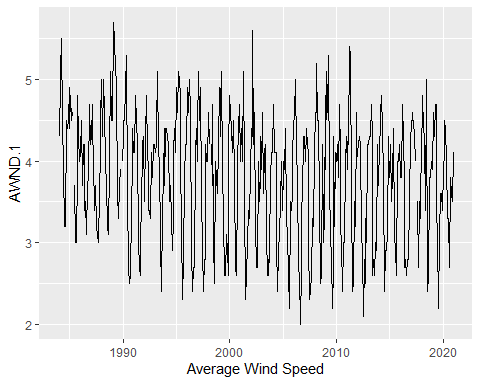
Many of the functions we will use require a time series object. You can transform your data in a time series using the function *ts()*.

ts\_wind\_data <- ts(wind\_data$AWND.1,frequency=12)

## Initial plots

Common plots for outlier detection are histograms and boxplots. Histograms will help you understand the shape and spread of the data and to identify any potential outliers. And boxplots will give more information on the spread of the data.

#using package ggplot2 to make histograms  
  
ggplot(wind\_data, aes(x = DATE, y = AWND.1)) +  
 geom\_line() +  
 xlab("Average Wind Speed")

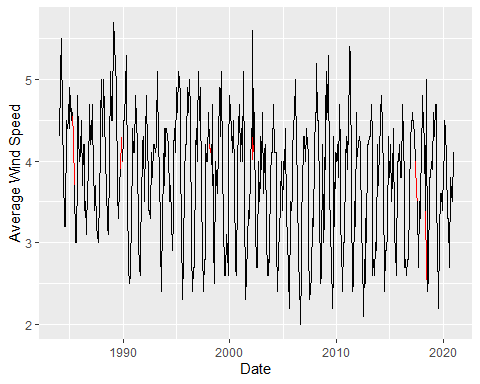
 ## Using pre-built function ts clean series

tsclean() from package forecast identify and replace outliers and missing values in a time series using linear interpolation.

clean\_wind\_data <- tsclean(ts\_wind\_data) #object should be a time series  
  
full\_wind\_data <- data\_frame(Date=wind\_data$DATE, Wind = wind\_data$AWND.1, Wind\_Clean = as.numeric(clean\_wind\_data))

## Warning: `data\_frame()` was deprecated in tibble 1.1.0.  
## Please use `tibble()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

ggplot(full\_wind\_data, ) +  
 geom\_line(aes(x = Date, y = Wind\_Clean), color = "red") +  
 geom\_line(aes(x = Date, y = Wind), color = "black") +  
 ylab("Average Wind Speed")



## Decomposing the time series

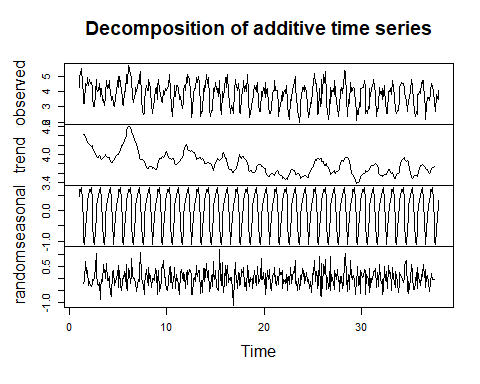
The stats package has a function called decompose(). This function only take time series object. As the name says the decompose function will decompose your time series into three components: trend, seasonal and random. This is similar to what we did in the previous script, but in a more automated way.

The random component is the time series without seasonal and trend component. Let’s try to identify outliers by looking at the random component only.

Additional info on decompose()

1. You have two options: alternative and multiplicative. Multiplicative models exhibit a change in frequency over time.
2. The trend is not a straight line because it uses a moving average method to detect trend.
3. The seasonal component of the time series is found by subtracting the trend component from the original data then grouping the results by month and averaging them.
4. The random component, also referred to as the noise component, is composed of all the leftover signal which is not explained by the combination of the trend and seasonal component.

#Ex1: Using R decompose function decompose the wind data in seasonal, trend and randamo  
decompose\_wind\_data=decompose(clean\_wind\_data)  
plot(decompose\_wind\_data)



#Ex2: Store the three component in separte vector  
wind\_random <- decompose\_wind\_data$random   
wind\_trend <- decompose\_wind\_data$trend  
wind\_seasonal <- decompose\_wind\_data$seasonal   
  
#Ex3: Create one data frame with all three vector and teh Date columns from wind\_data  
df\_wind\_decomposed <- data.frame("date"=full\_wind\_data$Date,wind\_trend,wind\_seasonal,wind\_random)  
  
#Ex4: Inspect the data frame using head() and tail()  
head(df\_wind\_decomposed,10)

## date wind\_trend wind\_seasonal wind\_random  
## 1 1984-01-01 NA 0.4466800 NA  
## 2 1984-02-01 NA 0.6860318 NA  
## 3 1984-03-01 NA 0.6240866 NA  
## 4 1984-04-01 NA 0.7419346 NA  
## 5 1984-05-01 NA 0.3404405 NA  
## 6 1984-06-01 NA -0.3960229 NA  
## 7 1984-07-01 4.420833 -0.9931381 -0.2276952  
## 8 1984-08-01 4.416667 -1.1079496 -0.1087170  
## 9 1984-09-01 4.383333 -0.6006580 0.7173246  
## 10 1984-10-01 4.320833 -0.2014682 0.2806348

tail(df\_wind\_decomposed,10)

## date wind\_trend wind\_seasonal wind\_random  
## 435 2020-03-01 3.687500 0.6240866 0.18841338  
## 436 2020-04-01 3.716667 0.7419346 -0.05860128  
## 437 2020-05-01 3.712500 0.3404405 -0.05294053  
## 438 2020-06-01 3.737500 -0.3960229 -0.04147706  
## 439 2020-07-01 NA -0.9931381 NA  
## 440 2020-08-01 NA -1.1079496 NA  
## 441 2020-09-01 NA -0.6006580 NA  
## 442 2020-10-01 NA -0.2014682 NA  
## 443 2020-11-01 NA 0.1278509 NA  
## 444 2020-12-01 NA 0.3322124 NA

# Discuss the results

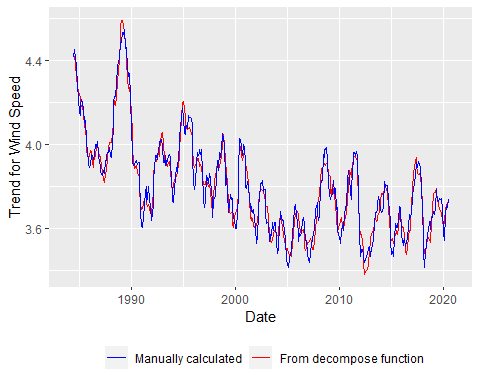
Discussion: Why the missing values??

## Coding challenge

#Ex5: Try to reproduce the trend component from decompose().  
order <- 6 #frequency = 12  
rep\_trend <- array(NA,length(clean\_wind\_data))  
for(t in (order+1):(length(clean\_wind\_data)-(order))){  
 rep\_trend[t] = mean(clean\_wind\_data[(t-order):(t+order)])  
}  
  
ggplot(data.frame(Date = as.Date(wind\_data$DATE),as.numeric(wind\_trend), as.numeric(rep\_trend))) +  
 geom\_line(aes(x = Date, y = wind\_trend, color = "Decompose")) +  
 geom\_line(aes(x = Date, y = rep\_trend, color = "Reproduced")) +  
 labs(color="") +  
 scale\_color\_manual(values = c("Reproduced" = "blue", "Decompose" = "red"),  
 labels=c("Manually calculated", "From decompose function")) +  
 theme(legend.position = "bottom") +  
 ylab(label="Trend for Wind Speed")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.

## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).



#Ex6: Try to reproduce the seasonal component from decompose().  
library(tidyverse)  
library(lubridate)  
  
#Start by eliminating the trend to calculate seasonal component  
orig\_minus\_trend <- clean\_wind\_data - wind\_trend  
orig\_minus\_trend <- data.frame(Date=as.Date(wind\_data$DATE),series=as.numeric(orig\_minus\_trend))  
  
orig\_minus\_trend\_by\_month<-  
 orig\_minus\_trend %>%   
 drop\_na(series) %>%   
 mutate(month=month(Date)) %>%   
 group\_by(month) %>%   
 summarize(rep\_seas=mean(series))   
  
orig\_minus\_trend\_by\_month <- cbind(orig\_minus\_trend\_by\_month,"wind\_seasonal" = wind\_seasonal[1:12])  
  
ggplot(orig\_minus\_trend\_by\_month,aes(x = month)) +  
 geom\_line(aes(y = rep\_seas,color="Reproduced")) +  
 geom\_line(aes(y = wind\_seasonal,color="Decompose")) +  
 labs(color="") +  
 scale\_color\_manual(values = c("Reproduced" = "blue", "Decompose" = "red"),  
 labels=c("Manually calculated", "From decompose function")) +  
 theme(legend.position = "bottom") +  
 ylab(label="Seasonal Comp for Wind Speed")

