[STAT 4540] HW-2

Michael Ghattas

2/9/2022

# Problem 1

let

Let

### (a)

Given the below points, we can conclude that there exists a unique and stationary solution for :

2. and

### (b)

Given that , thus the process is non-causal.

### (c)

Given that , thus the process is invertible.

### (d)

Thus: .

# Problem 2

### (a)

The big picture is filtering the estimated trend from the data through filtering out until we are left with nothing but residuals that are stationary.

### (b)

No, the shift by re-indexing should have not affect any significant change.

### (c)

The condition should not depend on the indexing choice, however the output would change if we redefined the indexing sequence.

# Problem 3

### (a)

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(gridExtra)  
library(ggplot2)  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.2

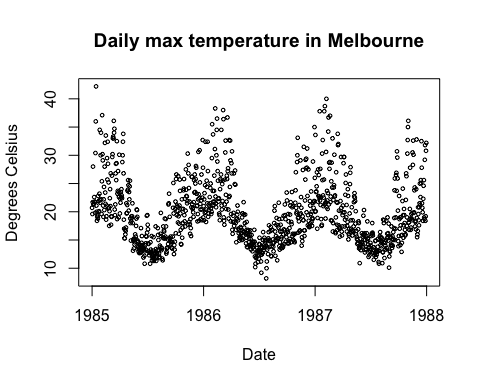
##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

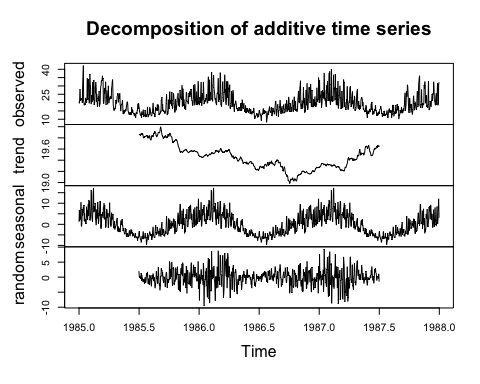
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

data <- load("/Users/Home/Documents/Michael\_Ghattas/School/CU\_Boulder/2022/Spring 2022/STAT - 4540/HW/2/DailyMaxMelbourne19851987.RData")  
dates = as.Date(dates,format = "%m/%d/%y")  
df = data.frame(dates, temp)  
plot(dates, temp, xlab = "Date", ylab = "Degrees Celsius", main = "Daily max temperature in Melbourne", pch = 01, cex = 0.5)



### (b)

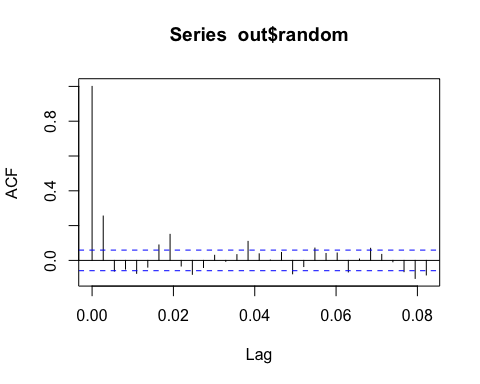
df.ts <- ts(data = df$temp, start = c(1985, 1), end = c(1987, 365), frequency = 365)  
out <- decompose(df.ts, type = "additive")  
plot(out)



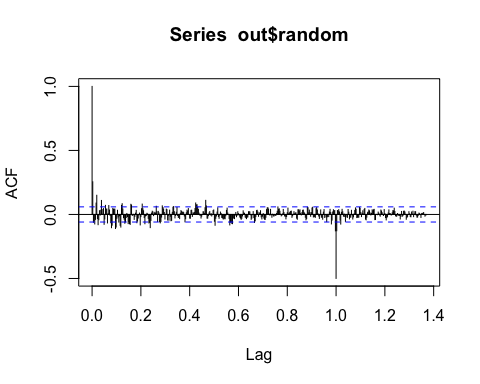
We can see that the decomposition did not work well, as the results are to noisy. An appropriate filter is needed prior to the decompose function.

### (c)

acf(out$random, na.action = na.pass)



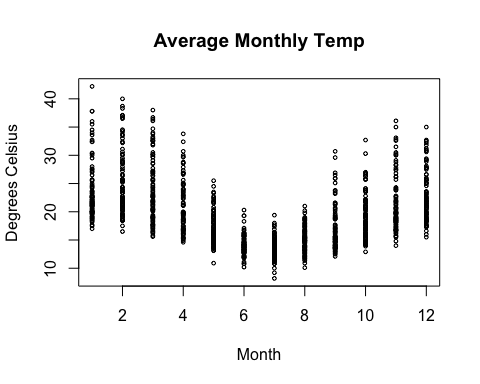
acf(out$random, na.action = na.pass, lag.max = 500)



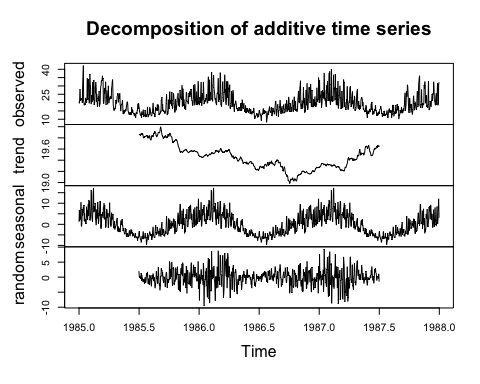
The results present moderate correlation, thus further filtering is needed to be able to identify white noise. A low-degree polynomial based regression model might be helpful.

### (d)

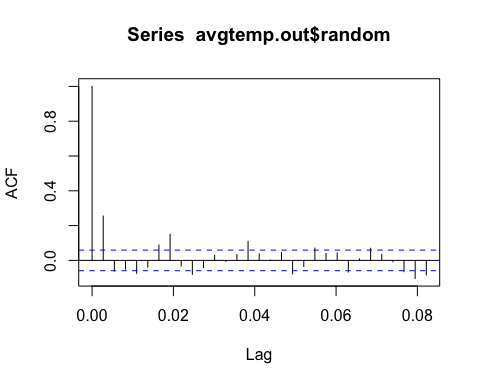
df$year <- year(df$dates)  
df$month <- month(df$dates)  
  
avgtemp <- aggregate(temp ~ year + month, df, mean)  
avgtemp$date = as.Date(paste(avgtemp$year, avgtemp$month, 01), "%y %m %d")  
avgtemp = arrange(avgtemp, date)  
  
avgtemp.ts <- ts(data = avgtemp$temp, start = c(1985, 1), end = c(1987, 12), frequency = 12)  
avgtemp.out <- decompose(df.ts, type = "additive")  
  
plot(df$month, temp, xlab = "Month", ylab = "Degrees Celsius", main = "Average Monthly Temp", pch = 01, cex = 0.5)



plot(avgtemp.out)



acf(avgtemp.out$random, na.action = na.pass)



From the data we can see a correlation between the month and average temperature. We can also see a relationship between the the average temperature of each month in relation to the previous month. Additional filtering is needed to extract trends, seasonality and noise.

# Problem 4

### (a)

Thus: satisfies an AR(p) stationary process.