Homework 9

Kenn Son, Hamid Suda, Vivian Truong

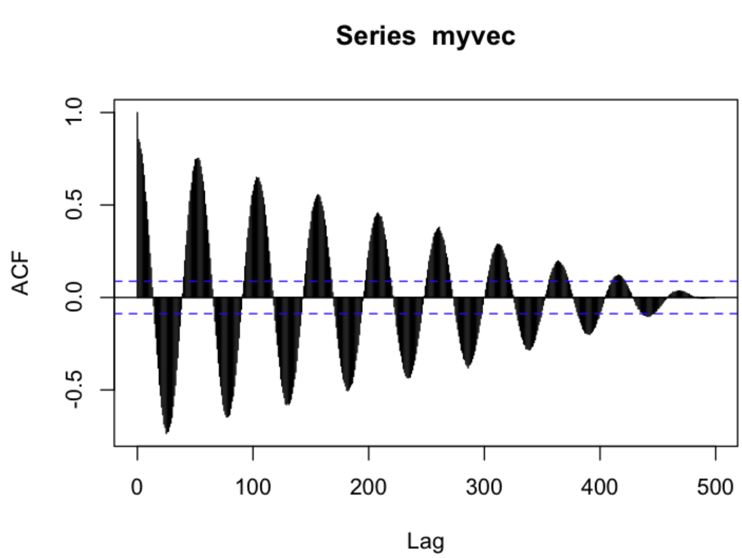
Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. You may work with students in both sections (375-01, -02). Only one person in the group should submit to Canvas.
Due: check on Canvas.

1. Load the "mystery" vector in file myvec.RData on Canvas (using `load("myvec.RData")`. Note that R allows you to store objects in its own machine-independent binary format instead of a text format such as .csv). Decompose the time series data into trend, seasonal, and random components.
Specifically, write R code to do the following:

a) Load the data. [show code]
load("/Users/owner/Downloads/myvec.RData")
> head(myvec)
[1] 16.06551 18.37622 18.42437 14.68903 16.74474 24.08552
> view(myvec[1:500])

b) Find the frequency of the seasonal component (Hint: use the autocorrelation plot. You must specify the lag.max parameter in acf() as the default is too small.) [show code and plot]

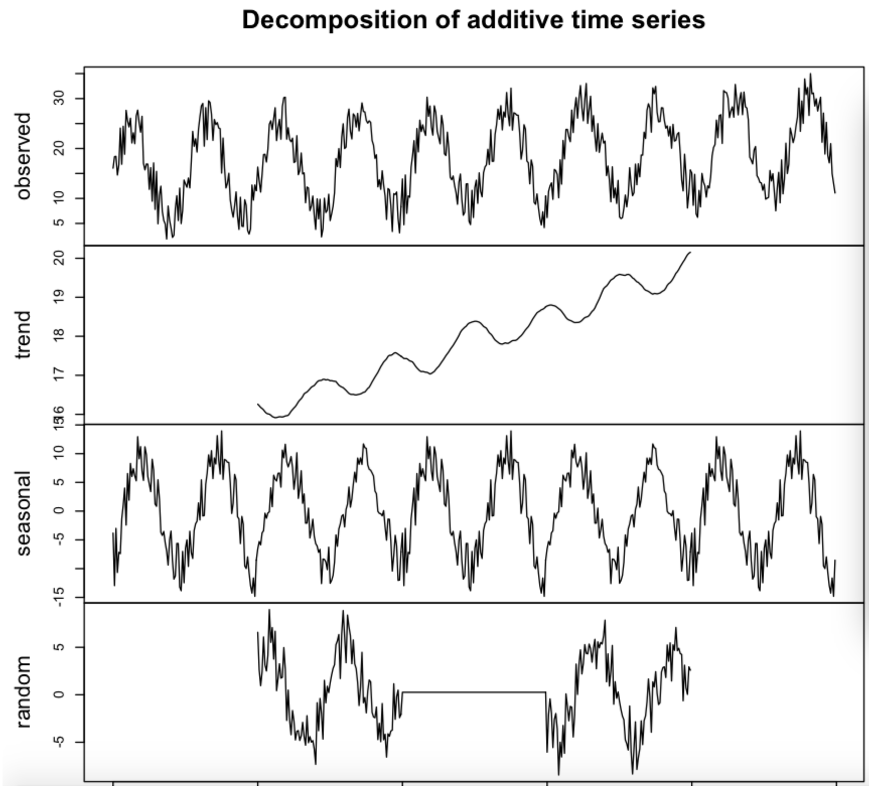
> acf(myvec,lag.max=500)



c) Convert to a `ts` object [show code]
> myvec.ts <- ts(myvec, frequency = 200)

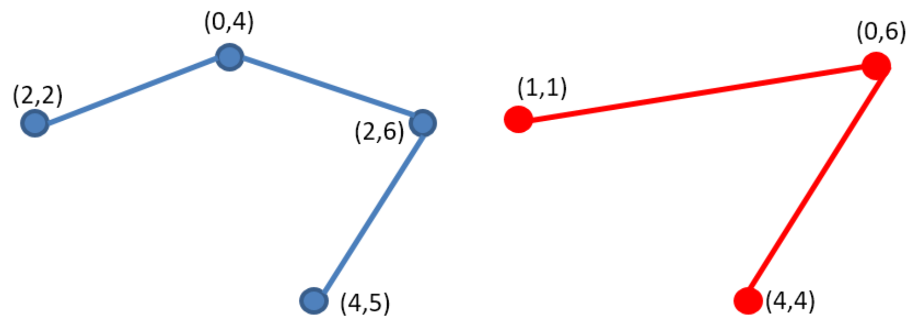
d) Decompose the ts object. Plot the output showing the trend, seasonal, random components. [show code and plot]

plot(decompose(myvec.ts))



2. (Same as classwork problem) Compute the Dynamic Time Warping distance between the two time series, A and B:

A= (2,2), (0,4), (2,6), (4,5)
B= (1,1), (0,6), (4,4)



Use squared Euclidean distance as the cost function:

$cost(A_i, B_j) = (A_{i,x} - B_{j,x})^2 + (A_{i,y} - B_{j,y})^2$.

(2-1)^2+(2-1)^2 = 2	(2-0)^2+(2-6)^2 = 20	(2-4)^2+(2-4)^2 = 8
(0-1)^2+(4-1)^2 = 10	(0-0)^2+(4-6)^2 = 4	(0-4)^2+(4-4)^2 = 16
(2-1)^2+(6-1)^2 = 26	(2-0)^2+(6-6)^2 = 4	(2-4)^2+(6-4)^2 = 8
(4-1)^2+(5-1)^2 = 25	(4-0)^2+(5-6)^2 =17	(4-4)^2+(5-4)^2 = 1

a) Show the cost matrix. This is partially complete below.

	B_1	B_2	B_3
A_1	2	20	8
A_2	10	4	16
A_3	26	4	8
A_4	25	17	1

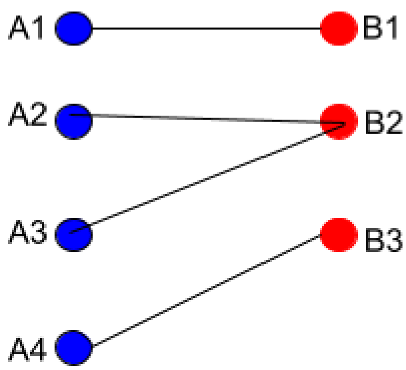
b) Show the DTW matrix. This is partially complete below.

B_1	B_2	B_3

A_1	2	22	30
A_2	12	6	22
A_3	38	10	14
A_4	63	27	11

c) The DTW distance between the two time-series is __**11**__.

d) Mark the optimal alignment between the two time-series in the diagram below.



3. a) Complete the R function below to compute the DTW distance between two time-series, v1 and v2, each containing 2D points and using the cost function as in Q2 above. So v1 and v2 will have two columns but different numbers of rows.

```
dtw <- function (A, B) {
  M <- nrow(A)
  N <- nrow(B)
  Cost <- matrix(0,M,N) # Initialize with zeros
  for (i in 1:M) {
    for (j in 1:N) {
        Cost[i,j] <- as.numeric((A[i,1] - B[j,1])^2 + (A[i,2] -
B[j,2])^2) # distance function
      }
  }
  C <- matrix(0,M,N) # Initialize with zeros
  C[1,1] <- Cost[1,1] # Initialize top left cell
  for (i in 2:M) { # Initialize first column
      C[i,1] <- C[i-1,1] + Cost[i,1]
  }
  for (j in 2:N) { # Initialize first row
      C[1,j] <- C[1,j-1] + Cost[1,j]
```

```
    }
    # Complete the main loop
        for(i in 1:M){
          for (j in 1:N){
            C[i,j]<- min(C[i-1,j],C[i,j-1],C[i-1,j-1]+Cost(i,j))
          }
    }
    return (C[M,N])
  }
```

b) Verify your answer to Q2 using the above function. [show code]

Hint: You can create the two input time-series as a two-column data.frame/tibble like so:

```
A <- tibble("x" = c(2, 0, 2, 4), "y" = c(2, 4, 6, 5))
```

4. You are given 5 time-series of 2D points (2 column tables) in CSV files: ts2.csv, ts3.csv, ts4.csv, ts5.csv, and tsX.csv. Your goal is to identify which of the time series, ts2-ts5, is most similar to the tsX time series using DTW.

a) Explain your approach in 2-3 sentences.

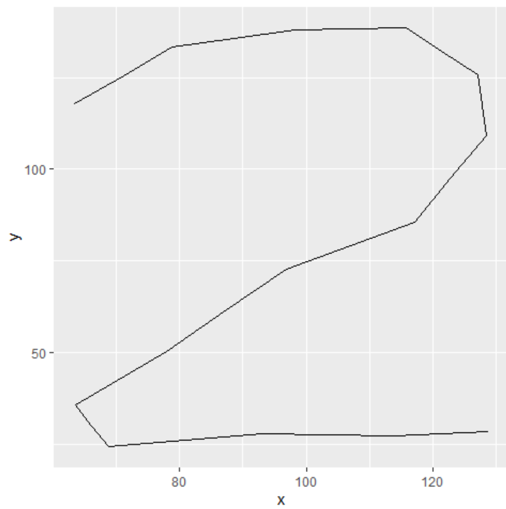
The goal is identify which of the time series are most similar to tsX in which I would use geom_path command to analyze each of the .csv. Then I would also calculate the 5 time series using DTW as a confirmation.

b) Show your R code

c) tsX is most similar to: _ts5___

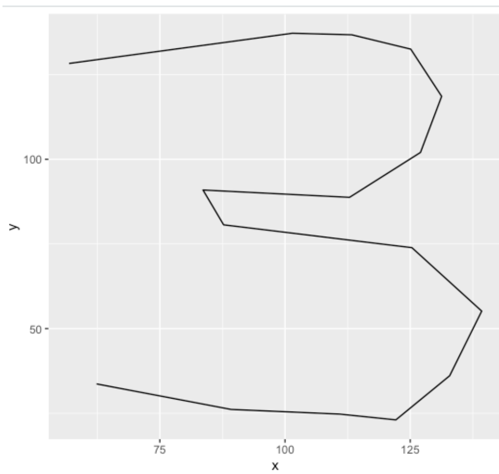
Hint: Use the DTW function from Q3. You can visualize the series of 2D points using geom_path(). For example, ts2:

```
>ts2 <- read_csv("ts2.csv")
>ts2
> m <- ggplot(ts2, aes(x,y))
> m +geom_path()
```



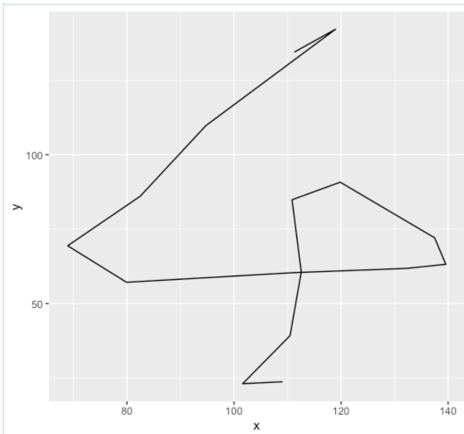
ts3:

```
>ts3 <- read_csv("ts3.csv")
>ts3
> m <- ggplot(ts3, aes(x,y))
> m +geom_path()
```

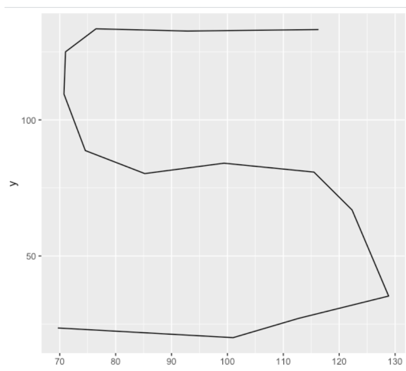


Ts4:

```
>ts4 <- read_csv("ts4.csv")
>ts4
> m <- ggplot(ts4, aes(x,y))
> m +geom_path()
```



Ts5:
```
>ts5 <- read_csv("ts5.csv")
>ts5
> m <- ggplot(ts5, aes(x,y))
> m +geom_path()
```



tsX:
```
>tsX <- read_csv("tsX.csv")
>tsX
> a <- ggplot(tsX, aes(x,y))
> a +geom_path()
```