Homework 7

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.

Bryce Lin, Serop Kelkelian

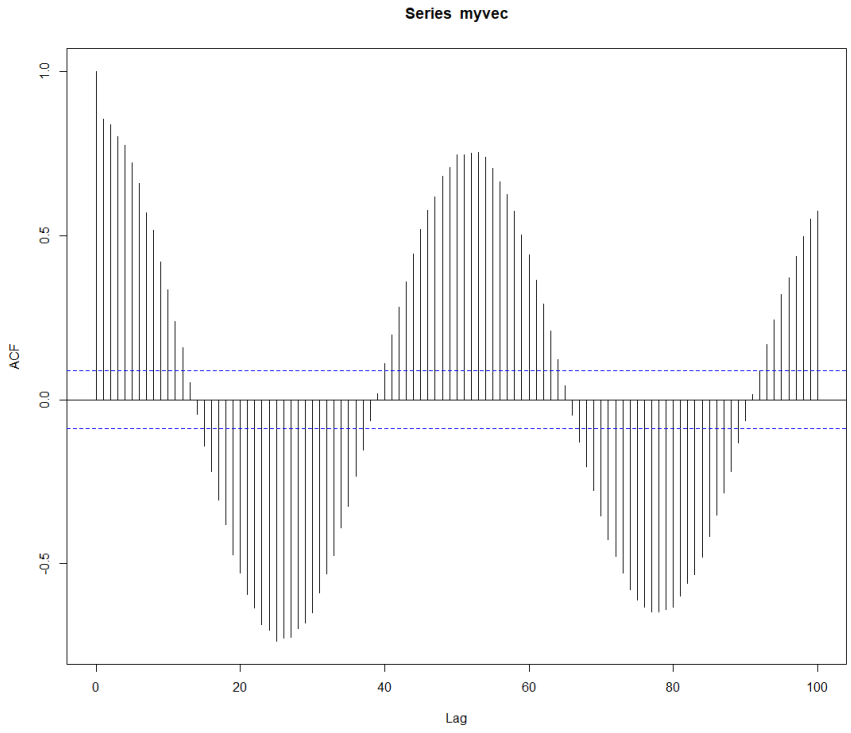
1. Load the "mystery" vector in file `myvec.RData` on Canvas under Datasets using `load("myvec.RData")`[1]. Decompose the time series data into trend, seasonal, and random components.
Specifically, write R code to do the following:

a) Load the data. [show code]

```
myvec <- get(load("myvec.rdata"))
```

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.) [code and plot]

```
acf(myvec, lag.max = 100)
```



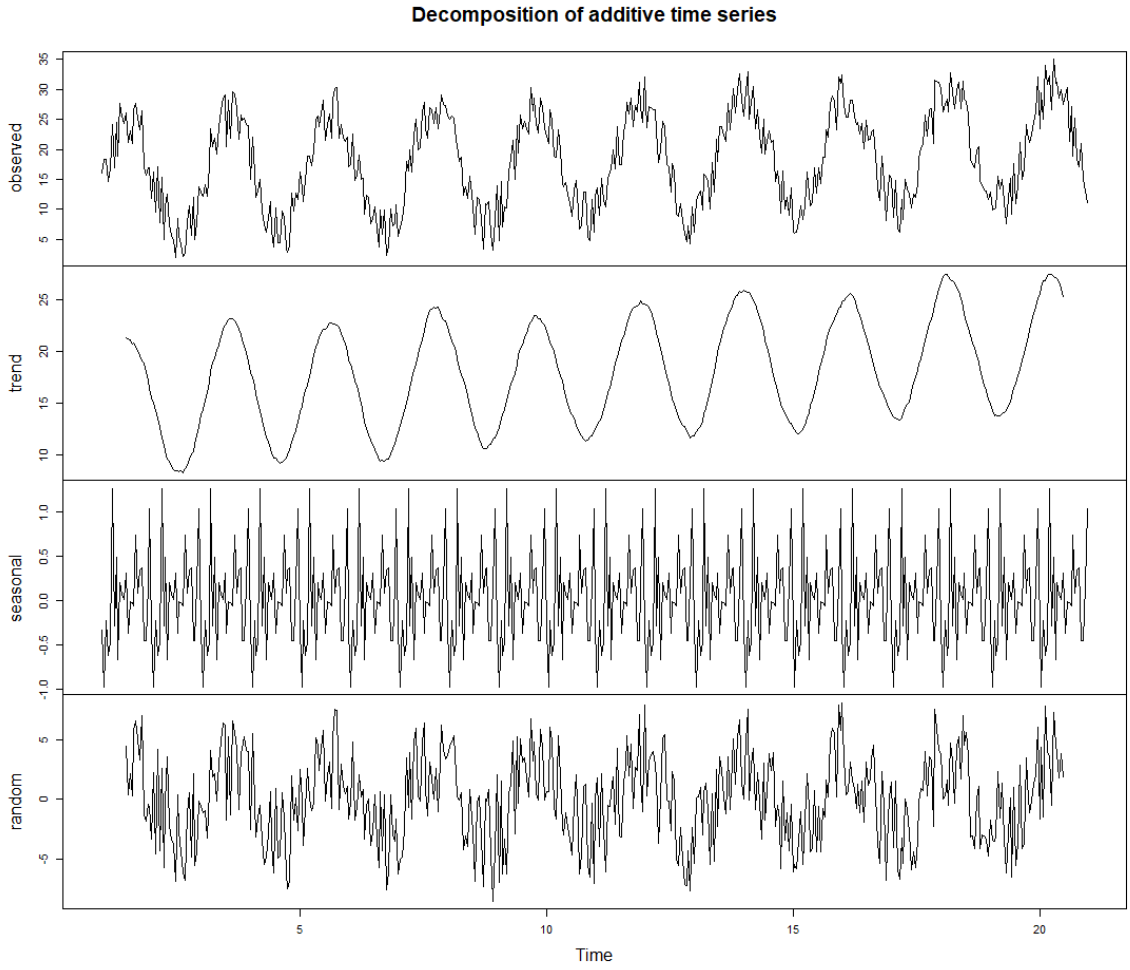
c) Convert to a `ts` object [code]

```
myvec.ts <- ts(myvec, frequency = 25)
```

[1] R allows you to store objects in its own machine-independent binary format, .RData, instead of a text format such as .csv

d) Decompose the ts object. Plot the output showing the trend, seasonal, random components. [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:

$cost(A_i, B_j) = (A_{i,1} - B_{j,1})^2 + (A_{i,2} - B_{j,2})^2$.

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

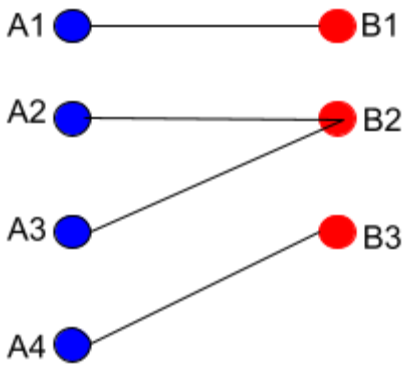
	B_1	B_2	B_3
A_1	2	20	8
A_2	10	4	16
A_3	26	4	8
A_4	25	17	1

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

2	22	30
12	6	22
38	10	14
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, A and B, each containing 2D points and using the cost function as in Q2 above. So A and B will have two columns but a varying number 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] # Value for top left cell
  for (i in 2:M) { # Values for first column
    C[i,1] <- C[i-1,1] + Cost[i,1]
  }
  for (j in 2:N) { # Values for first row
    C[1,j] <- C[1,j-1] + Cost[1,j]
  }
  #
  # Values for other rows and columns
  # TO BE COMPLETED
  #
  for (i in 2:M){
    for (j in 2: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. You can create the two input time-series as a two-column data.frame/tibble like so:

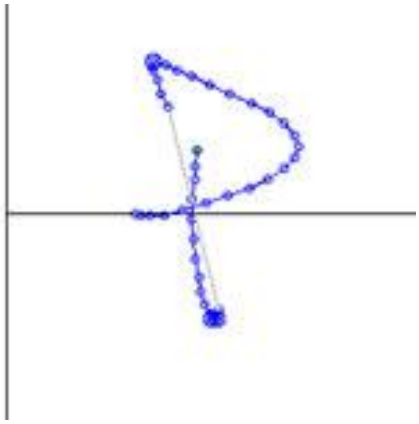
```
A <- tibble("x" = c(2, 0, 2, 4), "y" = c(2, 4, 6, 5))
```

[show code and output]

```
A <- tibble("x" = c(2, 0, 2, 4), "y" = c(2, 4, 6, 5))
B <- tibble("x" = c(1,0,4), "y" = c(1,6,4))
dtw(A,B)
```

```
> dtw(A,B)
[1] 11
```

4. You are given 4 time-series of 2D points (2 column tables) in CSV files: char1_A.csv, char1_E.csv, char1_M.csv, char1_O.csv, and char4_.csv (under Datasets module on Canvas). Each represents one of the English alphabet



characters A, E, M, and O as written on a tablet computer[2]. For instance, char1_A.csv, represents the character “A”. Your goal is to identify which character is represented by the 5th file (char4_.csv) using DTW and the cost function used in Q2 and Q3.

a) Explain your approach in 2-3 sentences.

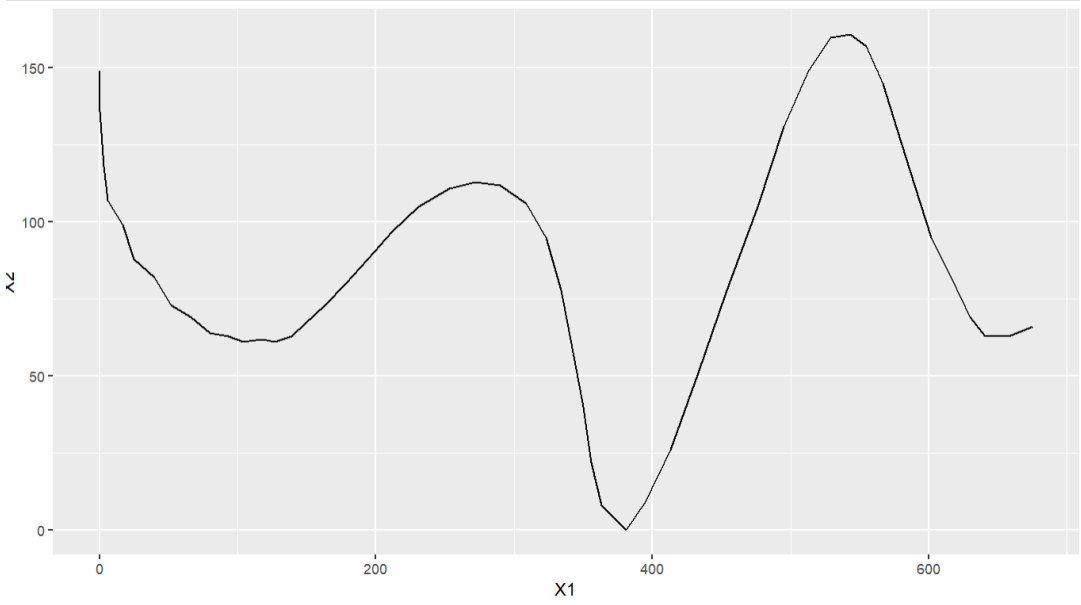
Import all the cvs file into the Rstudio and compare all known cases to the unknown cases to see which one have the lowest cost.

b) Show your R code and output

```
> A <- read.csv("char1_A.csv")
> E <- read.csv("char1_E.csv")
> M <- read.csv("char1_M.csv")
> O <- read.csv("char1_O.csv")
> unknown <- read.csv("char4_.csv")
> dtw(A,unknown)
[1] 3562468
> dtw(E,unknown)
[1] 2335604
> dtw(M,unknown)
[1] 979030
> dtw(O,unknown)
[1] 2089999
>
```

c) char4_.csv represents character: _M___

Hint: Use the DTW function from Q3. You can visualize the series of 2D points using geom_path() to check your answer.



[2] Data from UCI Machine Learning repository: https://archive.ics.uci.edu/ml/datasets/UJI+Pen+Characters