Table of Contents

Problem 1

Consider the following interaction with MATLAB.

```
data = readtable('Sample_data_xlsx.xlsx'); % import data
data(1:5,:) % print the first five rows
```

ans = 5×11 table

...

	state_code	state	gdp_2009	gdp_2010	gdp_2011	gdp_2012
1	'AK'	'Alaska'	44215	43472	44232	44732
2	'AL'	'Alabama'	149843	153839	155390	157272
3	'AR'	'Arkansas'	89776	92075	92684	93892
4	'AZ'	'Arizona'	221405	221016	224787	230641
5	'CA'	'California'	1667152	1672473	1692301	1751002

This data set describes the gross domestic products (GDP) and GDP growth rates of US states. The columns named `gdp_growth_2009`, `gdp_growth_2010, gdp_growth_2011 and gdp_growth_2012` show the GDP growth rates in 2009, 2010, 2011 and 2012, respectively.

1. Use logical operators to determine states that have positive growth rates in 2009.
2. Use logical operators to determine states that have negative growth rates in 2010, but positive growth rates in 2009.
3. Which states have positive growth rates in all years? Which states have negative growth rates in all years? (via Logical indexing)
4. Are there any states that have zero growth rates? (via Logical indexing)

```
% Solution to Part 1
```

Problem 2

Consider the following interaction with MATLAB.

```
A=zeros(5,8);
```

Let `A(i,j)` be the `(i,j)`th element of `A,` where `i=1,2,,..5` and `j=1,2,...8`. Use a `for` loop to replace each element of `A` in the following way.

- If `i>j` set `A(i,j)=4*i-2j.`
- If `i<=j` set `A(i,j)=i^2-3j.`

Also, do the same task with a `while` loop.

```
% Solution
% Using for loop
% Using while loop
```

Problem 3

The natural exponential function can be expressed by $e^x = \lim_{k\to\infty} \sum_{n=0}^{k} \frac{x^n}{n!}$. We would like to compute e^2. Use both `for` and `while` loops to compute the sum of series when `(a) k=5`, `(b) k=15` and `(c) k=25.` Compare your results with `exp(2)`. (Use `format long`.)

```
% Solution
```

Problem 4

Throughout this problem, consider the following setting:

```
rng('default') % Set the random number generator to the default seed
```

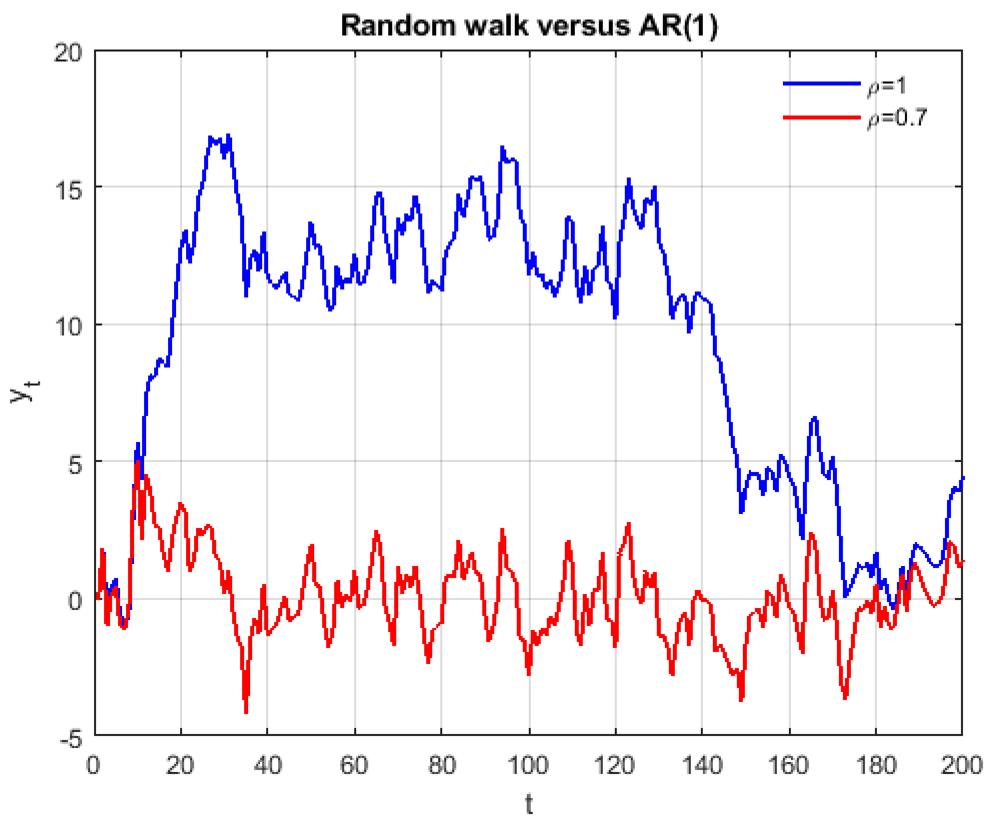
A first order autoregressive model (AR(1)) can be defined in the following way.

$$y_t = \rho y_{t-1} + u_t, \quad t = 1, 2, \ldots, T,$$

where y_t denotes the value of y at time t, ρ is a parameter and u_t is a random variable that has a standard normal distribution. When $\rho = 1$, the model is called the *random walk* model. Assume that $y_0 = 0$ (when `t=0`), and T=200. Use a `for` loop to generate 200 observations according this model when $\rho = 1$ and $\rho = 0.7$. Plot your resulting observations in both cases. Your code snippet should generate the figure given below. Recall that the `normrnd` function can be used to generate random numbers from the standard normal distribution

```
% Solution
```

yt and plot



Problem 5

The letter grades and their numerical versions are given in the following table.

Letter	Grade Points
A+	4
A	4.0
A-	3.7
B+	3.3
B	3.0
B-	2.7
C+	2.3
C	2.0
C-	1.7
D+	1.3
D	1.0
F	0

Write an `if-elseif-else-end` structure that converts the letter grades to their equivalent number of grade points. Your program should ask the user to enter a letter grade. Ensure that your program generates an appropriate error message if the user enters an invalid letter grade.

```
% Solution
```

Problem 6

Consider **Problem 5**. Instead of `if-elseif-else-end` structure, use the `switch-case` structure to do the same task.

```
% Solution
```

Problem 7

Consider the following interaction with MATLAB

```
rng('default');
x = 0:.05:10;
n=length(x);
y = cos(x)+0.3*normrnd(0,1,[1,n]);
```

The vector `y` is the sum of `cos(x)` and a noise term represented by `0.3*normrnd(0,1[1,n]).` We would like to use the smoothing techniques to get smoothed versions of y. The simplest smoothing algorithm is the *rectangular boxcar* or *unweighted sliding-average smooth*; it simply replaces each point in the signal with the average of *m* adjacent points, where *m* is a positive integer called the *smooth width*. For example, for a 3-point smooth (*m* = 3)

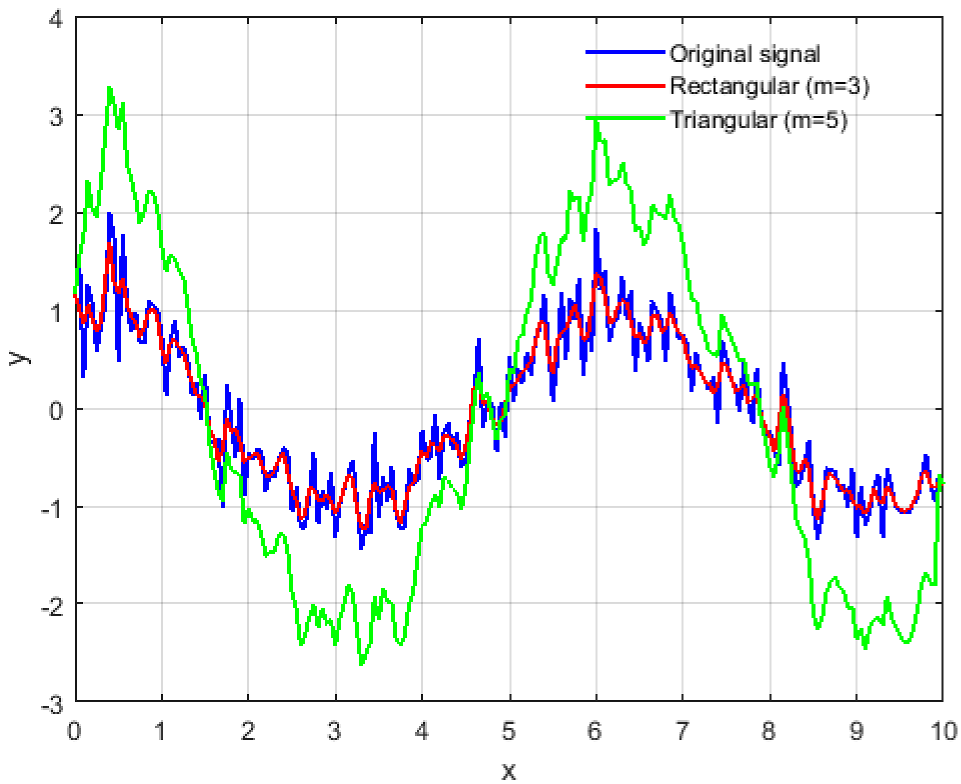
$$s(i) = \frac{y(i-1) + y(i) + y(i+1)}{3}$$

for $i = 2, 3, \ldots, n-1$, where $s(i)$ the *i*th point in the smoothed signal. The *triangular smooth* is like the rectangular smooth, above, except that it implements a *weighted* smoothing function. For a 5-point smooth (*m* = 5)

$$s(i) = \frac{y(i-2) + 2y(i-1) + 3y(i) + 2y(i+1) + y(i+2)}{5}$$

for $i = 3, 4 \ldots, n-2$. Write a program that produces the smoothed versions of y according to both methods. Make a plot that display the noisy and smoothed signals. Your code snippet should generate the following figure.

```
% Solution rectangular, triangular and plot
```



Problem 8

Consider the following interaction with MATLAB.

```
rng('default');
v=chi2rnd(3,[1,10]); % generate 10 random numbers from chi-square distribution
```

Using conditional statements and loops, write a program that rearranges the elements of v in order from the largest to the smallest. Do not use MATLAB's built-in function `sort`.

```
% Solution
```