Practical Assignment: Jupyter Notebook and Python

(40% of overall course assessment. Total score of 40 marks)

Complete the follow tasks in a Jupyter notebook and submit your work via the submission point on PolyMall.

Markdown and Documentation (8 marks)

It is useful to provide documentation in codes to aid understanding and using markdown aids in the legibility of these content

Create (using Markdown)

1. Header Level 1 title. Name it “Assignment 1”.
2. Table consisting of 3 column “admin no.”, “name” and “email contact”. Fill in with the relevant row data.
3. Header Level 2 title. Name it “Experience”.
4. List of your programming experiences. You can list down “Nil” as a list item if not applicable.
5. Header Level 3 title. Name it “Code”.
6. Create a code section to display a string assignment to a variable [str = “Hello”] and printing of the string variable [print (str)].
7. Header Level 4 title. Name it “Equation”.
8. Math equation $\sqrt{4} \neq 1$

   The following is an example of the final documentation.

Assignment 1

admin no	name	email contact
123456X	John Smith	john.smith@email.com

Experience

My programming experience

- C++
- Java
- Nil

Code

```
str = "Hello"
print (str)
```

Equation

$$\sqrt{4} \neq 1$$

Python

Python Part 1 (8 marks):

Generate two sets of arrays representing the value of dice thrown by two dice players. For each throw, the larger number win the set. If the number thrown is the same, it is a draw.

The two players automatically play for 5 times (throw the dice 5 times).

Create a python code snippet that compare the value of dice thrown, counts the number of sets won by each player and print the outcome of the winner and the difference in number of wins.

Note that the two players may draw if they both win the same number of sets.

Iterate your code snippet 5 times and print each game's result.

The value of dice thrown is expected to be random and should differ from the sample output. The following sample output of the five games is formatted to be displayed within two columns to aid in viewing within the document and need not be followed in your actual code.

Sample output

```
Player 1
[1 5 5 3 1]
Player 2
[3 6 5 1 2]
Player 2 wins by 2 game
```

```
Player 1
[4 5 4 5 6]
Player 2
[5 6 3 2 2]
Player 1 wins by 1 game
```

```
Player 1
[5 6 4 3 2]
Player 2
[2 3 2 4 1]
Player 1 wins by 3 game
```

```
Player 1
[2 4 2 4 6]
Player 2
[4 2 4 6 2]
Player 2 wins by 1 game
```

```
Player 1
[1 5 5 5 3]
Player 2
[2 1 5 6 1]
It is a draw with 2 win each.
```

Python Part 2 (8 marks):

Follow the general steps to obtain the sample output.

Step 1: Store the following tables of data into three DataFrames,

Step 2: Add a column to store the month of the sales

Step 3: Concatenate the three DataFrames into a single DataFrame.

January

Quantity Sold	Product	Unit Price
500	Apples	$1.00
600	Oranges	$2.00

February

Quantity Sold	Product	Unit Price
900	Apples	$0.80
300	Bananas	$0.50

March

Quantity Sold	Product	Unit Price
200	Apples	$0.80
100	Bananas	$1.00

Sample output

	Quantity Sold	Product	Unit Price	Month
0	500	Apples	1	Jan
1	600	Oranges	2	Jan
2	900	Apples	0.8	Feb
3	300	Bananas	0.5	Feb
4	200	Apples	0.8	Mar
5	100	Bananas	1.0	Mar

Python Part 3 (8 marks):

Given the following sets of weight and height for 20 adults, use a scatter plot in matplotlib to display the data as follows.

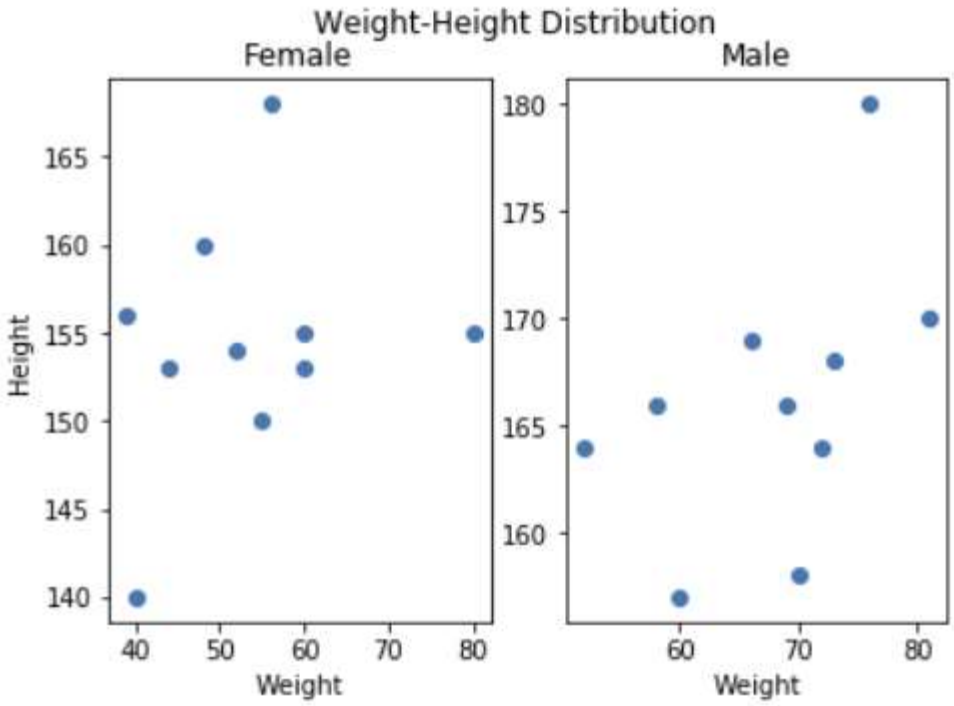
Female adults= ([56,60,55,52,40,60,80,39,44,48], [168,155,150,154,140,153,155,156,153,160])

Female										
Weight	56	60	55	52	40	60	80	39	44	48
Height	168	155	150	154	140	153	155	156	153	160

Male adults = ([60,70,73,72,81,52,76,58,66,69], [157,158,168,164,170,164,180,166,169,166])

Male										
Weight	60	70	73	72	81	52	76	58	66	69
Height	157	158	168	164	170	164	180	166	169	166

Sample output



Python Part 4 (8 marks):

Create a function to determine if a given passcode is well-formed. A passcode is well-formed if

1. It is in the format <part-1>#<part-2>
2. The <part-1> consists of only alphanumeric characters or dot '.'
3. The dot '.' cannot be the first or last character in <part-1> and it cannot appear consecutively
4. The <part-2> only consists of alphanumeric characters and no character is to be repeated within <part-2>.

Do not use REGEX to solve this problem.

Input

cases = ['abc#abc', '1#patch', 'a.b#cd9', '12to13#timesup', 'a.b.c#987', '.ab#123','a..b#123', 'ab.#123','ab.c#1.23','1234.abc', 'ab12#pass', 'ab-12#past','ab12#pas$']

Sample output

```
abc#abc is valid
1#patch is valid
a.b#cd9 is valid
12to13#timesup is valid
a.b.c#987 is valid
.ab#123 is not valid
a..b#123 is not valid
ab.#123 is not valid
ab.c#1.23 is not valid
1234.abc is not valid
ab12#pass is not valid
ab-12#past is not valid
ab12#pas$ is not valid
```