SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

ICT233
Data Programming

Tutor-Marked Assignment

January 2022 Presentation

TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth 24 % of the final mark for ICT233, Data Programming.

The cut-off date for this assignment is **Monday, 14 Mar 2022, 2355 hours.**

Note to Students:

You are to include the following particulars in your submission: Course Code, Title of the TMA, SUSS PI No., Your Name, and Submission Date.

Please refer to the additional TMA submission instructions :
https://canvas.suss.edu.sg/courses/45712/discussion_topics/247324

Answer all questions. (Total 100 marks)

Question 1 (60 marks)

Objectives:

- Understand dataset with data scientist mind-set.
- Understand and design computation logic and routines in Python.
- Assess use of Python only and Python data structures to perform extract, load, and transformation operations.
- Assess use of Pandas dataframe to perform extract, load, transformation and calculation operations.
- Structure code in appropriate methods (functions), looping and conditions.
- Conduct visualization in an appropriate way.

There are 6 synthetic datasets, which capture tracing data of a virus spread within a population.

1. *f0_f1.csv*: contains f0 to f1 relationship.

   Each row (**trace_id**, **f0**, **f1**) shows a person ID at the **f0** column infected another person ID at the **f1** column in the cluster indicated by **trace_id**.

2. *f1_f2.csv*: contains f1 to f2 relationship.
3. *f2_f3.csv*: contains f2 to f3 relationship.
4. *f3_f4.csv*: contains f3 to f4 relationship.
5. *f4_f5.csv*: contains f4 to f5 relationship.
6. *people.csv*: contains **person ID** and **name** of people in the population.

(a) Each cluster is represented by a unique **trace ID** across all 5 datasets: *f0_f1.csv*, *f1_f2.csv*, ..., and *f4_f5.csv*. To use **dataframe** to compute the number of unique people for every cluster (**trace ID**).

(10 marks)

(b) Visualize the people count per cluster dataframe outputted by **1a** on a **histogram** with x axis showing the people count and y axis showing the number of clusters having the people count. Analyse the output and share **ONE** insight which you may draw from the diagram.

(10 marks)

(c) The *people.csv* contains the population of **ALL** people under the contact tracing collection. Use **dataframe** to find all people who are **NOT** in any cluster (potentially negative or no close contact).

(10 marks)

(d) Use **dataframe** and design a function which takes a **person ID** and a **trace ID** as its parameters and returns a dataframe showing all tracing paths passing through the specified person ID. With the help of this function, it provides information on all people who potentially infected the specified person directly/indirectly and whom were potentially infected by the specified person directly/indirectly.
For example, when calling the function with **person ID** = *PERSON_0000000067* and **trace ID** = *TRACE_PERSON_0000000379*, a sample of the returned dataframe is showed in Figure 1 below.

(10 marks)

	f0	f1	f2	f3	f4	f5
0	PERSON_0000000379	PERSON_0000000218	PERSON_0000000054	PERSON_0000000889	PERSON_0000000640	PERSON_0000000067
1	PERSON_0000000379	PERSON_0000000218	PERSON_0000000054	PERSON_0000000889	PERSON_0000000067	PERSON_0000000209
2	PERSON_0000000379	PERSON_0000000218	PERSON_0000000054	PERSON_0000000889	PERSON_0000000067	PERSON_0000000456
3	PERSON_0000000379	PERSON_0000000740	PERSON_0000000054	PERSON_0000000889	PERSON_0000000640	PERSON_0000000067
4	PERSON_0000000379	PERSON_0000000740	PERSON_0000000054	PERSON_0000000889	PERSON_0000000067	PERSON_0000000209
5	PERSON_0000000379	PERSON_0000000740	PERSON_0000000054	PERSON_0000000889	PERSON_0000000067	PERSON_0000000456

Figure 1: Sample output of Q1(d)

(e) Use **dataframe** and design a function which takes in 3 inputs: a **trace ID**, a **"from"** person ID and a **"to"** person ID, and returns all tracing paths starting from the **"from"** person ID and ending at the **"to"** person ID in the specified cluster by the **trace ID**. In other words, the function answers whether the **"from"** person infected directly/indirectly the **"to"** person in the specified cluster.

- **Direct** relationship: there is a direct edge to connect the two persons, for example, A <-> B.
- **Indirect** relationship: To connect a person A to a person C, it goes through a non-empty set of persons, for example, A <-> B1 <-> B2 ... <-> C.

(10 marks)

(f) Apply/Call the function defined in **Q1(d)(i)** with the following parameters: **person ID** = PERSON_0000000067 and **trace ID** = TRACE_PERSON_0000000379. Then follow the steps in the link (https://www.datacamp.com/community/tutorials/networkx-python-graph-tutorial) to use **networkx** to visualize the returned cluster (sample diagram is showed in Figure 2)

(10 marks)

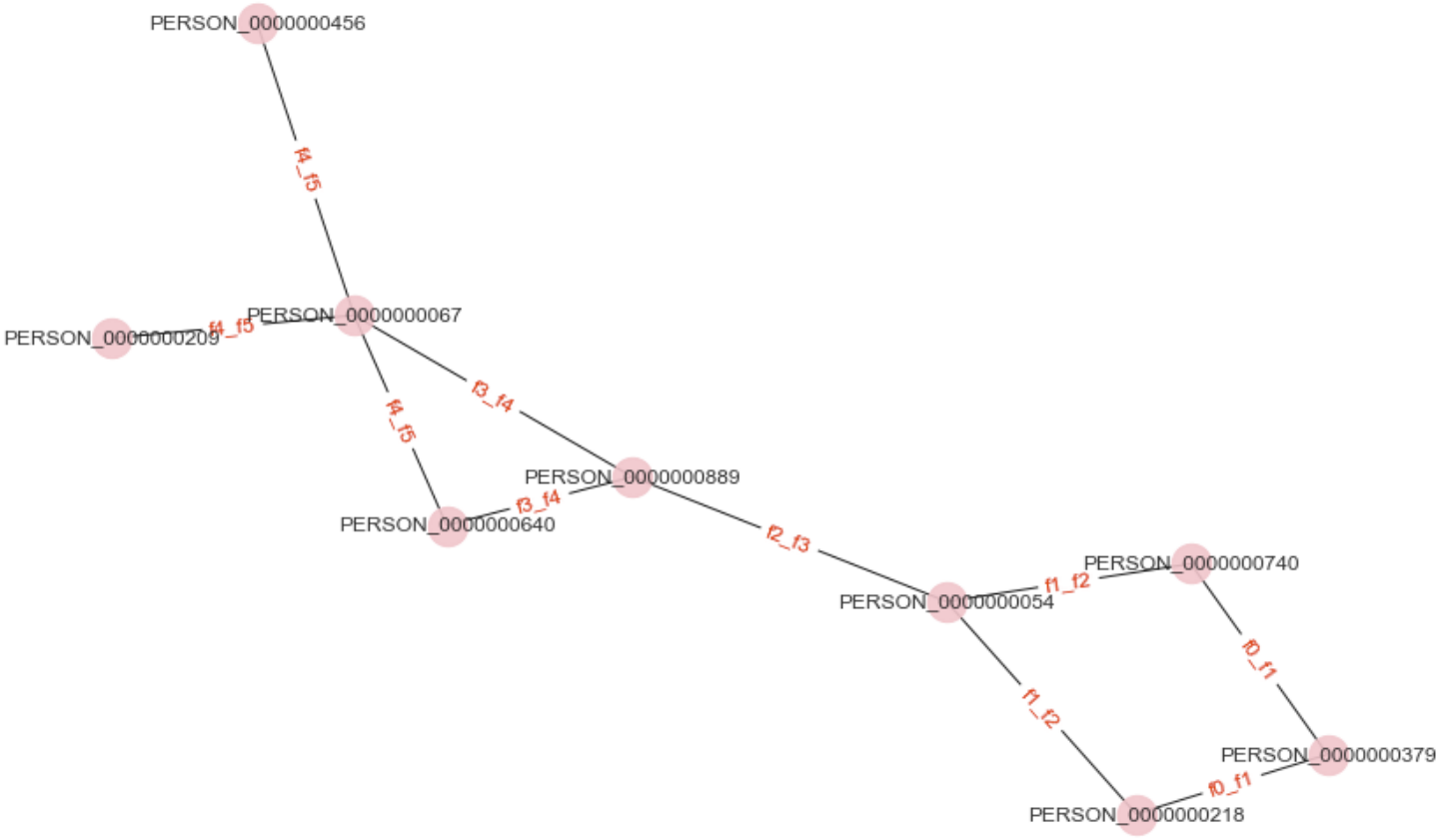

Figure 2: Sample output of Q1(f)

Question 2 (40 marks)

Objectives:

- Understand dataset with data scientist mind-set.
- Design computation logic and routines in Python.
- Conduct visualization in an appropriate way.
- Perform simple exploratory data analysis.
- Assess the design and use of database ORM and methods to perform extract, load, transformation and calculation operations.

There are 2 synthetic datasets:

1. The *people.csv* has 2 fields: **person_id** and **name**.
2. The *acquaintance.csv* has 2 fields: **from** & **to**, which indicates 2 person IDs who know each other.

(a) Use **sqlalchemy ORM** to define and store data of 2 entities **Person** and **Acquaintance**, which can be loaded from *people.csv* and *acquaintance.csv* correspondingly. Note to define the correct relationship between the 2 entities.

(8 marks)

(b) Compose necessary queries and define a function which takes a **person ID** as its parameter and implement the function using **sqlalchemy ORM** to find all direct acquaintances' names of the given person ID.

- **Direct** relationship: there is a direct edge to connect the two persons, for example, A <-> B in the *acquaintance.csv.*

(8 marks)

(c) Develop programme to perform the following tasks:

(i) Use **sqlalchemy ORM** to count the number of acquaintances per person.

(5 marks)

(ii) Draw a boxplot to display the data distribution of the number of acquaintances per person.

(3 marks)

(d) Use **sqlalchemy ORM** to find the **names** of people that having the most number of acquaintances.

(8 marks)

(e) Use **sqlalchemy ORM** to find all groups of **THREE (3)** **distinct** people who all know each other.

(8 marks)

---- END OF ASSIGNMENT ----