```
%%% Q.3: Of the 4 numeric data below, do two or more appear to be correlated:
% number of 'switched on' and 'switched off' statuses in a network-state
% how many complete paths arise from a network-state
% how long paths arising from a network-state tend to be on average
% how many distinct FPs appear across all paths arising from a network-state

% Function to create table bearing information about network-nodes individual
statuses, number of active & inactive network-nodes,
% average (complete) path lengths, number of paths, & number of FPs appearing in
graph-node paths
function insightsTable = createInsightsTable(states_dict, binaryStrings, FPs, G,
distinctFPsDictionary)

graphNodes = values(states_dict); % Store graph-node IDs

networkStates = binaryStrings; % Store network-states

FPsforState = values(distinctFPsDictionary); % Extract the number of FPs that
paths starting from current network-state end in

% Find individual network-node statuses and number of active and inactive
network-nodes
for inBinaryStrings = 1:length(binaryStrings) % Iterate through all binary
network-states

    currentState = binaryStrings(inBinaryStrings, :); % Extract current network-
state

    stateArray = arrayfun(@str2double, currentState); % Row array of current
network-state

    numOneStatuses = length(stateArray(stateArray == 1)); % Count number of '1'
statuses in current network-state

    numZeroStatuses = 6 - numOneStatuses; % Count number of '0' statuses in
current network-state

    for inStateArray = 1:length(stateArray) % Iterate through all values in
current network-state

        if inStateArray == 1 % For node status of node 1 (TGFβ)

            TGFbeta(inBinaryStrings, 1) = stateArray(inStateArray); % Store
status value

        elseif inStateArray == 2 % For node status of node 2 (miRNA200)

            miRNA200(inBinaryStrings, 1) = stateArray(inStateArray); % Store
status value

        elseif inStateArray == 3 % For node status of node 3 (Snail1)
```

```
            Snail1(inBinaryStrings, 1) = stateArray(inStateArray); % Store
status value

        elseif inStateArray == 4 % For node status of node 4 (Zeb1)

            Zeb1(inBinaryStrings, 1) = stateArray(inStateArray); % Store status
value

        elseif inStateArray == 5 % For node status of node 5 (Ovol2)

            Ovol2(inBinaryStrings, 1) = stateArray(inStateArray); % Store status
value

        elseif inStateArray == 6 % For node status of node 6 (miRNA34a)

            miRNA34a(inBinaryStrings, 1) = stateArray(inStateArray); % Store
status value

        end

    end

    numOnes(inBinaryStrings, 1) = numOneStatuses; % Store current network-
state's number of 1 among its network-node statuses

    numZeros(inBinaryStrings, 1) = numZeroStatuses; % Store current network-
state's number of 0 among its network-node statuses

    % Find average length of paths starting with current network-state and
ending in all FPs
    sumAllPathLengths = 0; % Initialize and reset value of sum of all (complete)
path lengths

    numberAllPaths = 0; % Initialize and reset value of number of all (complete)
path lengths

    currentStatePaths = {}; % Initialize and reset cell to store all complete
paths of network-state

    for inFPs = 1:length(FPs) % Iterate through row vector of FPs

        currentFP = FPs(inFPs); % Extract current FP

        currentStatePaths{inFPs, 1} = allpaths(G, graphNodes(inBinaryStrings),
currentFP); % Store all paths ending in FPs

        pathLengthsArray = []; % Initialize and reset row vector for next
iteration

        for inCurrentStatePaths = 1:length(currentStatePaths{inFPs, 1}) %
Iterate through all paths for current FP

            pathLengthsArray(end + 1) = length(currentStatePaths{inFPs, 1}
{inCurrentStatePaths}); % Store length of each (complete) path in array
```

```
        end

        sumFPpathLengths = sum(pathLengthsArray); % Sum of all paths ending in
current FP

        numberFPpaths = length(pathLengthsArray); % Number of paths ending in
current FP

        sumAllPathLengths = sumAllPathLengths + sumFPpathLengths; % Update the
sum of all (complete) paths starting from current network-state

        numberAllPaths = numberAllPaths + numberFPpaths; % Update the number of
all (complete) paths starting from current network-state

    end

    avgPathLength(inBinaryStrings, 1) = round(sumAllPathLengths /
numberAllPaths, 1); % Store value of average (complete) path lengths starting
from current

    % Find number of paths starting from network-state
    numberStatePaths(inBinaryStrings, 1) = numberAllPaths;

    % Find (complete) paths starting from current network-state end in how many
FPs
    numDistinctFPs(inBinaryStrings, 1) = FPsforState(inBinaryStrings); % Store
number of distinct FPs appearing at the end of paths starting from current
network-state

end

% Create and display insights table
insightsTable = table(graphNodes, networkStates, TGFbeta, miRNA200, Snail1,
Zeb1, Ovol2, ...
    miRNA34a, numOnes, numZeros, avgPathLength, numberStatePaths,
numDistinctFPs); % Create insights table

disp('Insights Table:');
disp(insightsTable);

end

% Execute function to obtain insights table
insightsTable = createInsightsTable(statesDictionary, binaryStatesChar, FPs, G,
distinctFPsDictionary);
```

Output:

```
Insights Table:

<64 x 13 table>
```