Probability Tree

* Place a series of probabilities in to Matrix [A].
* Each column is one set and should add up to 100%.
* If one set has more possible outcomes than another, fill empty rows with 0.
* Run the program
* The program will step through and find each possible combination of probabilities laid out in Matrix [B].
* The final column of Matrix [B] will be the total and each previous column will show which value it used to from each set to make the total.

Example:

A company makes dolls in wrapped boxes.

The dolls have these attributes:

60% Female, 40% Male

50% Green shirt, 30% Yellow shirt, 20% Red shirt

90% Blue pants, 10% Black pants

|  |  |  |
| --- | --- | --- |
| Matrix [A] – Input | | |
| 0.6 | 0.5 | 0.9 |
| 0.4 | 0.3 | 0.1 |
| 0 | 0.2 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Matrix [B] – Output | | | |
| 1 | 1 | 1 | 0.27 |
| 1 | 1 | 2 | 0.03 |
| 1 | 2 | 1 | 0.162 |
| 1 | 2 | 2 | 0.018 |
| 1 | 3 | 1 | 0.108 |
| 1 | 3 | 2 | 0.012 |
| 2 | 1 | 1 | 0.18 |
| 2 | 1 | 2 | 0.02 |
| 2 | 2 | 1 | 0.108 |
| 2 | 2 | 2 | 0.012 |
| 2 | 3 | 1 | 0.072 |
| 2 | 3 | 2 | 0.008 |

Row 1 shows the most likely outcome

(Female, Green Shirt, Blue Pants 27%)

Row 12 shows the least likely outcome

(Male, Red Shirt, Black Pants 0.8%)

12 total rows = 12 possible combinations

Variables

Matrix [A] – Input

Matrix [B] – Output

Matric [C] – Used for counting through each set of

probabilities.

L1 – Used to store the dimensions of Matrix [A]

A – Number of rows in Matrix [A]

B – Number of columns in Matrix [A]

G – Used to calculate the expected number of outcomes

P – Used to step through the number of possible outcomes

X – Used to step through rows

Y – Used to step through columns

Step-by-step

1) Stores the dimensions of Matrix [A] to L1

2) Stores the number of rows from Matrix [A] to A

3) Stores the number of columns from Matrix [A] to B

4) Creates counting matric [C] with B number of columns

and 2 rows

5) Fills Matrix [C] with 1s to facilitate counting

6) Resets G to 1 for counting

7) Resets P to 1 for counting

8) Loop that counts Y from 1 to B (columns)

9) Resets X to 1 for counting

10) Loop that counts X through to A (rows)

11) Looks for a 0 value to stop counting of rows

12) Then

13) Make X large enough to exit the loop from line 10

14) Else

15) Store X as the max number of values in row 1, column Y

of Matrix [C]

16) Count X up by 1

17) End of loop from line 11

18) End of loop from line 10 (rows)

19) Multiply G by the number of values in the column

20) End loop from line 8 (columns)

21) Creates Matrix [B] with G rows and B+1 columns

22) Fill Matrix [B] with 1s for later multiplication

23) Loop while the counting number is less than the max in

column 1 of Matrix [C]

24) Loop that counts Y from 1 to B (columns)

25) Multiply the current value from the current column to

the final total.

26) Store which value was used in Matrix [B]

27) End loop from line 24

28) Decrease Y so it can be used to step backwards through

the columns

29) Increase the value counter in the last column by 1

30) Loop steps through value columns as long as each value

counter needs to be increased

31) resets the value counter in the current column

32) Moves the column focus

33) adds 1 to current column

34) End loop from line 30

35) Increases outcome counter (Matrix [B] row)

36) End loop from line 23