We are trying to create a graph using NetworkX and plot it using Matplotlib.

```
import networkx as nx
import matplotlib.pyplot as plt
# create an undirected graph
G = nx.Graph()
G.add_node("ba", probs={'ba_y': 0.2, 'ba_n': 0.8})
G.add_node("ab", probs={'ab_y': 0.1, 'ab_n': 0.9})
G.add_node("fb", probs={'fb_y': 0.3, 'fb_n': 0.7})
G.add_node("bd", probs={'ba_y_bd_y': 0.7, 'ba_y_bd_n': 0.3 , 'ba_n_bd_y': 0.3, 'ba_n_bd
G.add_node("nc", probs={'ab_y_fb_y_nc_y': 0.75, 'ab_y_fb_n_nc_y': 0.4,'ab_n_fb_y_nc_y':
G.add_node("bm", probs={'bd_y_bm_y': 0.9, 'bd_y_bm_n': 0.1, 'bd_n_bm_y':0.1,'bd_n_bm_n'
G.add_node("bf", probs={'bd_y_nc_y_bf_y': 0.95, 'bd_y_nc_n_bf_y': 0.85, 'bd_n_nc_y_bf_y
G.add_node("no", probs={'no_y': 0.05, 'no_n': 0.95})
G.add_node("ng", probs={'ng_y': 0.05, 'ng_n': 0.95})
G.add_node("flb", probs={'flb_y': 0.1, 'flb_n': 0.9})
G.add_node("sb", probs={'sb_y': 0.1, 'sb_n': 0.9})
G.add_node("l", probs={'l_y_bf_y': 0.9, 'l_n_bf_y': 0.3, 'l_y_bf_n':0.1, 'l_n_bf_n':0.7
G.add_node("ol", probs={'bf_y_no_y_ol_y': 0.9, 'bf_y_no_n_ol_y': 0.7,'bf_n_no_y_ol_y':0
G.add_node("gg", probs={'bf_y_ng_y_gg_y':0.95, 'bf_y_ng_n_gg_y':0.4,'bf_y_ng_n_gg_y':0.
G.add_node("cs", probs={'bf_y_no_y_ng_y_fb_y_sb_y_cs_n':0.9,'bf_y_no_y_ng_y_fb_y_sb_n_c
                        'bf_y_no_y_ng_n_fb_y_sb_y_cs_n':0.9,'bf_y_no_y_ng_n_fb_y_sb_n_c
                        'bf_y_no_n_ng_y_fb_y_sb_y_cs_n':0.9,'bf_y_no_n_ng_y_fb_y_sb_n_c
                        'bf_y_no_n_ng_n_fb_y_sb_y_cs_n':0.9,'bf_y_no_n_ng_n_fb_y_sb_n_c
                        'bf_n_no_y_ng_y_fb_y_sb_y_cs_n':0.9,'bf_n_no_y_ng_y_fb_y_sb_n_c
                        'bf_n_no_y_ng_n_fb_y_sb_y_cs_n':0.9,'bf_n_no_y_ng_n_fb_y_sb_n_c
                        'bf_n_no_n_ng_y_fb_y_sb_y_cs_n':0.9,'bf_n_no_n_ng_y_fb_y_sb_n_c
                        'bf_n_no_n_ng_n_fb_y_sb_y_cs_n':0.9,'bf_n_no_n_ng_n_fb_y_sb_n_c
'bf_y_no_y_ng_y_fb_y_sb_y_cs_y': 0.1,
'bf_y_no_y_ng_y_fb_y_sb_n_cs_y': 0.1,
'bf_y_no_y_ng_y_fb_n_sb_y_cs_y': 0.1,
'bf_y_no_y_ng_y_fb_n_sb_n_cs_y': 0.1,
'bf_y_no_y_ng_n_fb_y_sb_y_cs_y': 0.1,
'bf_y_no_y_ng_n_fb_y_sb_n_cs_y': 0.1,
'bf_y_no_y_ng_n_fb_n_sb_y_cs_y': 0.1,
'bf_y_no_y_ng_n_fb_n_sb_n_cs_y': 0.1,
'bf_y_no_n_ng_y_fb_y_sb_y_cs_y': 0.1,
'bf_y_no_n_ng_y_fb_y_sb_n_cs_y': 0.1,
'bf_y_no_n_ng_y_fb_n_sb_y_cs_y': 0.1,
'bf_y_no_n_ng_y_fb_n_sb_n_cs_y': 0.1,
'bf_y_no_n_ng_n_fb_y_sb_y_cs_y': 0.1,
'bf_y_no_n_ng_n_fb_y_sb_n_cs_y': 0.1,
'bf_y_no_n_ng_n_fb_n_sb_y_cs_y': 0.1,
'bf_y_no_n_ng_n_fb_n_sb_n_cs_y': 0.1,
'bf_n_no_y_ng_y_fb_y_sb_y_cs_y': 0.1,
'bf_n_no_y_ng_y_fb_y_sb_n_cs_y': 0.1,
'bf_n_no_y_ng_y_fb_n_sb_y_cs_y': 0.1,
'bf_n_no_y_ng_y_fb_n_sb_n_cs_y': 0.1,
'bf_n_no_y_ng_n_fb_y_sb_y_cs_y': 0.1,
'bf_n_no_y_ng_n_fb_y_sb_n_cs_y': 0.1,
'bf_n_no_y_ng_n_fb_n_sb_y_cs_y': 0.1,
'bf_n_no_y_ng_n_fb_n_sb_n_cs_y': 0.1,
```

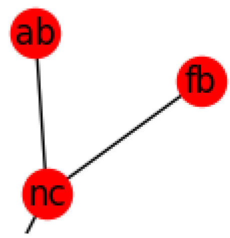
```
'bf_n_no_n_ng_y_fb_y_sb_y_cs_y': 0.1,
'bf_n_no_n_ng_y_fb_y_sb_n_cs_y': 0.1,
'bf_n_no_n_ng_y_fb_n_sb_y_cs_y': 0.1,
'bf_n_no_n_ng_y_fb_n_sb_n_cs_y': 0.1,
'bf_n_no_n_ng_n_fb_y_sb_y_cs_y': 0.1,
'bf_n_no_n_ng_n_fb_y_sb_n_cs_y': 0.1,
'bf_n_no_n_ng_n_fb_n_sb_y_cs_y': 0.1,
'bf_n_no_n_ng_n_fb_n_sb_n_cs_y': 0.9
}
)
G.add_node("d",probs={'no_y_dl_y':0.95,'no_n_dl_y':0.3,'no_y_dl_n':0.05,'no_n_d_n':0.7}

G.add_edge("ba", "bd")
G.add_edge("bd", "bm")
G.add_edge("bd", "bf")
G.add_edge("nc", "bf")
G.add_edge("ab", "nc")
G.add_edge("fb", "nc")
G.add_edge("bf", "l")
G.add_edge("bf", "ol")
G.add_edge("bf", "gg")
G.add_edge("bf", "cs")
G.add_edge("no", "ol")
G.add_edge("no", "cs")
G.add_edge("no", "d")
G.add_edge("ng", "cs")
G.add_edge("flb", "cs")
G.add_edge("sb", "cs")

node_color = "#FF0000"

# draw the graph using Matplotlib with node color
nx.draw(G, with_labels=True, node_color=node_color)
plt.show()

```



Below code calculates the conditional probability of an event "cs" being false given that events "ab" and "fb" are true, based on given probabilities for various variables.

```
# Assigning probabilities
P_nc_given_bd = 0.9
P_bd = 0.01
P_l_given_bf = 0.99
P_ol_given_bf_no = 0.9
P_gg_given_bf_ng = 0.9
P_cs_given_bf_no_ng_flb_sb = 0.1


total_sum = 0
count = 0

# Generating all possible combinations of free variables
for nc in [True, False]:
    for bf in [True, False]:
        for l in [True, False]:
            for ol in [True, False]:
                for gg in [True, False]:
                    for no in [True, False]:
                        for ng in [True, False]:
                            for flb in [True, False]:
                                for sb in [True, False]:
                                    # Calculating P(bf|nc,bd)
                                    P_bf_given_nc_bd = P_nc_given_bd if nc and sb else

                                    # Calculating joint probability
                                    joint_prob = (1 - P_cs_given_bf_no_ng_flb_sb) if no
                                                 P_bf_given_nc_bd * P_l_given_bf * P_ol
                                                 P_bf_given_nc_bd * P_l_given_bf * P_ol
                                                 P_bf_given_nc_bd * P_l_given_bf * P_ol
                                    total_sum += joint_prob
                                    count += 1

# Calculating conditional probability
P_cs_given_ab_fb = total_sum / count

# Printing the result
print("P(-cs|+ab,+fb) =", P_cs_given_ab_fb)

```

```
P(-cs|+ab,+fb) = 0.3475707609374998
```

R3

We can use the sum-product rule of probability to derive the probability of events "-cs" and "+ab" from the joint probability of events "-cs", "+ab", and "+fb". The sum-product rule allows us to eliminate the "+fb" variable in a similar way as in the previous calculation (referred to as R2).

The formula for calculating the probability of events "-cs" and "+ab" is as follows:

P(-cs, +ab) = ∑fb P(-cs, +ab, +fb)

This formula involves summing over the "+fb" variable and the free variables "nc" and "bd", and using conditional probabilities to eliminate the "+fb" and "+ab" variables.

The final formula after eliminating the "+fb" and "+ab" variables is as follows:

P(-cs, +ab) = ∑fb P(-cs | +ab, +fb) × P(+fb | +ab) × P(+ab) = ∑fb P(-cs | +ab, +fb) × P(+fb | +ab) × ∑nc, bd P(+ab | +nc, +bd) × P(+nc) × P(+bd)

In this formula, we are summing over the "+fb" variable and using conditional probabilities to calculate the probability of "-cs" given "+ab" and "+fb", the probability of "+fb" given "+ab", and the probability of "+ab" given "+nc" and "+bd". We also need to consider the probabilities of "+nc" and "+bd" as free variables in the calculation.

R3 - way 2

To eliminate the variable for fanbelt broken (fb) from the probability equation P(-cs, +ab, fb), we can sum over all possible values of fb, including fb_y (fanbelt broken) and fb_n (fanbelt not broken). The respective values for fb_y and fb_n are 0.3 and 0.7.

Using the sum over fb, we can express P(-cs, +ab) as follows: P(-cs, +ab) = ∑fb P(-cs|+ab, fb) P(+ab, fb)

By substituting the given values, we can eliminate the fb variable from the equation.

Alternatively, we can expand the equation similar to R2, and consider only the probability values from the table for the case of not charging (nc) when alternator is broken (ab) and fanbelt is not broken (fb_n). The probability value for this case is given as 0.4.

The parents of not charging (nc) are fb and ab (fanbelt broken and alternator broken), so the probability of not charging depends on both fb and ab. We can substitute the value of 0.4 in place of not charging, thereby eliminating the fb variable.

The simplified equation becomes: P(-cs, +ab) = ∑l, ol, gg, no,fb, ng, flb, sb P(+bf|(+nc, +bd)) × P(l|+bf) × P(ol|+bf, no) × P(gg|+bf, ng) × P(-cs|+bf, no,fb, ng, flb, sb)

R4

To derive P(-cs, +fb) from P(-cs, +ab, +fb), we can use the following formula:

P(-cs, +fb) = Σ(ab) P(-cs, +ab, +fb)

Here, we sum over all possible values of ab to eliminate the variable from the joint probability. This is similar to the method used in R2, where we summed over the bf variable to eliminate it from the joint probability.

Using Bayes' rule, we can also express the above formula in terms of conditional probabilities:

P(-cs, +fb) = Σ(ab) P(-cs | +ab, +fb) P(+ab | +fb)

Here, we use the fact that P(-cs, +ab, +fb) = P(-cs | +ab, +fb) P(+ab | +fb) P(+fb). We eliminate the +fb term by summing over all possible values of ab.

R4 way 2

To eliminate the variable for alternator broken (ab) from the probability equation P(-cs, +fb, ab), we can sum over all possible values of ab, including ab_y (alternator broken) and ab_n (alternator not broken). The respective values for ab_y and ab_n are 0.1 and 0.9.

Using the sum over ab, we can express P(-cs, +fb) as follows: P(-cs, +fb) = ∑ab P(-cs|+fb, ab) P(+fb, ab)

By substituting the given values, we can eliminate the ab variable from the equation.

Alternatively, we can expand the equation similar to R2, and consider only the probability values from the table for the case of not charging (nc) when fanbelt is broken (fb_y) and alternator is not broken (ab_n). The probability value for this case is given as 0.6.

The parents of not charging (nc) are fb and ab (fanbelt broken and alternator broken), so the probability of not charging depends on both fb and ab. We can substitute the value of 0.6 in place of not charging, thereby eliminating the ab variable.

The simplified equation becomes: P(-cs, +fb) = ∑l, ol, gg, no,fb, ng, flb, sb P(+bf|(+nc, +bd)) × P(l|+bf) × P(ol|+bf, no) × P(gg|+bf, ng) × P(-cs|+bf, no,fb, ng, flb, sb)

R5

If we are provided with actual numerical values for P(+bf|+bd) and P(+bf|+nc), we can compare them to determine the relative importance of the battery being dead and not charging in causing the battery to go flat.

If P(+bf|+bd) is significantly greater than P(+bf|+nc), it suggests that the battery being dead (+bd) is a more influential factor in causing the battery to go flat. On the other hand, if P(+bf|+nc) is considerably larger than P(+bf|+bd), it indicates that not charging (+nc) is more important in causing the battery to go flat.

However, if the values of P(+bf|+bd) and P(+bf|+nc) are similar, it implies that both factors may have comparable importance in determining whether the battery will go flat, and additional information may be required to draw a conclusive inference about which factor is more significant.

Careful analysis of the numerical values and consideration of the specific context of the problem are crucial in making meaningful interpretations about the relative importance of different factors in causing a particular outcome. Expertise in statistics and domain knowledge can aid in accurately interpreting the results.

Thank you

```
1
```

✓ 0s completed at 8:14 PM