6.S091: Problem Set 1

Hyewon Jeong
hyewonj@mit.edu

Problem 1: Interventions and Adjustment
1. Preliminaries
(a) What is the entailed distribution $\mathbb{P}_\chi$ of the SCM M?
From the information in the problem, we can draw the interventional augmented graph $\tilde{\mathcal{G}}$ as below:

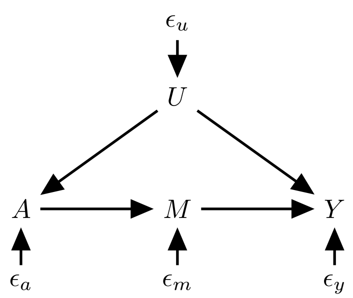

Figure 1. Interventional Augmented Graph $\tilde{\mathcal{G}}$.

Then we can calculate the entailed distribution:

$$\begin{aligned}\mathbb{P}_\chi(U=u,A=a,M=m,Y=y) &= \mathbb{P}_\chi(U=u)\mathbb{P}_\chi(A=a|U=u)\mathbb{P}_\chi(M=m|A=a)\mathbb{P}_\chi(Y=y|M=m,U=u)\\ &= Ber(U;0.5)\times Ber(A;\frac{U}{4})\times Ber(M;0.5+0.1A)\times Ber(Y;\frac{M}{2}+\frac{U}{4})\end{aligned}$$

(b) Based on the code run result of Figure 4, $\mathbb{P}_\chi(Y=1) = 0.38125$

$\mathbb{P}_\chi(Y=1|M=0,A=0) = 0.10714285714285714$

$\mathbb{P}_\chi(Y=1|M=0,A=1) = 0.25$

2. Interventional
(c) For the interventional distribution with the do operation do(A=1), we get:

$$\begin{aligned}\mathbb{P}_\chi(U=u,A=1,M=m,Y=y|do(A=1)) &= \mathbb{P}_\chi(U=u)\mathbb{P}_\chi(A=a|A=1)\mathbb{P}_\chi(M=m|A=1)\mathbb{P}_\chi(Y=y|M=m,U=u)\\ &= Ber(U;0.5)\times 1\times Ber(M;0.5+0.1)\times Ber(Y;\frac{M}{2}+\frac{U}{4})\\ &= Ber(U;0.5)\times Ber(M;0.6)\times Ber(Y;\frac{M}{2}+\frac{U}{4}) \qquad (1)\\ & \qquad (2)\end{aligned}$$

Likewise, $\mathbb{P}_\chi(U=u,A=1,M=m,Y=y|do(A=0)) = Ber(U;0.5)\times Ber(M;0.5)\times Ber(Y;\frac{M}{2}+\frac{U}{4})$.

(d) Using the equation 1, we can calculate the respective interventional probability as below:
$\mathbb{P}_\chi(Y=1|do(A=1) = 0.425$
$\mathbb{P}_\chi(Y=0|do(A=1)) = 0.575$
$\mathbb{P}_\chi(Y=1|do(A=0)) = 0.375$
$\mathbb{P}_\chi(Y=0|do(A=0)) = 0.625$

Backdoor Adjustment (e)

$$
\begin{aligned}
\mathbb{P}_\chi(Y=1|do(A=1)) &= \sum_{u\in\{0,1\}} \mathbb{P}_\chi(Y=1|U=u,A=1)P(U=u) \\
&= \mathbb{P}_\chi(Y=1|U=1,A=1)\mathbb{P}_\chi(U=1) + \mathbb{P}_\chi(Y=1|U=0,A=1)\mathbb{P}_\chi(U=0) \\
&= 0.5499999999999999 * 0.5 + \text{unavailable} * 0.5
\end{aligned}
$$

For the case of $\mathbb{P}_\chi(Y=1|do(A=1)$ and $\mathbb{P}_\chi(Y=0|do(A=1)$, we cannot find out whether it matches with the probability from (d) as some of the probabilities $(\mathbb{P}_\chi(Y=1,U=0,A=1), \mathbb{P}_\chi(U=0,A=1)$, are zero and unavailable to calculate $\mathbb{P}_\chi(Y=1|U=0,A=1)$.

$$
\begin{aligned}
\mathbb{P}_\chi(Y=0|do(A=1)) &= \sum_{u\in\{0,1\}} \mathbb{P}_\chi(Y=0|U=u,A=1)P(U=u) \\
&= \mathbb{P}_\chi(Y=0|U=1,A=1)\mathbb{P}_\chi(U=1) + \mathbb{P}_\chi(Y=0|U=0,A=1)\mathbb{P}_\chi(U=0) \\
&= 0.45000000000000007 * 0.5 + \text{unavailable} * 0.5
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}_\chi(Y=1|do(A=0)) &= \sum_{u\in\{0,1\}} \mathbb{P}_\chi(Y=1|U=u,A=0)P(U=u) \\
&= \mathbb{P}_\chi(Y=1|U=1,A=0)\mathbb{P}_\chi(U=1) + \mathbb{P}_\chi(Y=1|U=0,A=0)\mathbb{P}_\chi(U=0) \\
&= 0.5 \times 0.5 + 0.25 \times 0.5 = 0.375
\end{aligned}
$$

which matches with the probability from (d) where $\mathbb{P}_\chi(Y=1|do(A=0)) = 0.375$.

Likewise for the case of $\mathbb{P}_\chi(Y=0|do(A=0)$ it also matches with the result from above $\mathbb{P}_\chi(Y=0|do(A=0)) = 0.625$.

$$
\begin{aligned}
\mathbb{P}_\chi(Y=0|do(A=0)) &= \sum_{u\in\{0,1\}} \mathbb{P}_\chi(Y=0|U=u,A=0)P(U=u) \\
&= \mathbb{P}_\chi(Y=0|U=1,A=0)\mathbb{P}_\chi(U=1) + \mathbb{P}_\chi(Y=0|U=0,A=0)\mathbb{P}_\chi(U=0) \\
&= 0.5 \times 0.5 + 0.75 \times 0.5 = 0.625
\end{aligned}
$$

Frontdoor Adjustment (f)
We can show from the equation below that using front door adjustment, we can calculate the probability of $\mathbb{P}_\chi(Y=0|do(A=0))$ matching with the one that we calculated from (d).

$$
\begin{aligned}
\mathbb{P}_\chi(Y=1|do(A=1)) &= \sum_{m\in\{0,1\}} \mathbb{P}_\chi(M=m|A=1) \sum_{a'\in\{0,1\}} \mathbb{P}_\chi(Y=1|M=m,A=a')\mathbb{P}_\chi(A=a') \\
&= \mathbb{P}_\chi(M=1|A=1) \sum_{a'\in\{0,1\}} P(Y=1|M=1,A=a')\mathbb{P}_\chi(A=a') \\
&+ \mathbb{P}_\chi(M=0|A=1) \sum_{a'\in\{0,1\}} P(Y=1|M=0,A=a')\mathbb{P}_\chi(A=a') \\
&= 0.6 \times (0.6071428571428571 * 0.875 + 0.75 * 0.125) \\
&+ 0.4 \times (0.10714285714285714 * 0.875 + 0.25 * 0.125) \\
&= 0.425
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}_{\chi}(Y=0|do(A=1)) &= \sum_{m\in\{0,1\}} \mathbb{P}_{\chi}(M=m|A=1) \sum_{a'\in\{0,1\}} \mathbb{P}_{\chi}(Y=0|M=m,A=a')\mathbb{P}_{\chi}(A=a') \\
&= \mathbb{P}_{\chi}(M=1|A=1) \sum_{a'\in\{0,1\}} P(Y=0|M=1,A=a')\mathbb{P}_{\chi}(A=a') \\
&+ \mathbb{P}_{\chi}(M=0|A=1) \sum_{a'\in\{0,1\}} P(Y=0|M=0,A=a')\mathbb{P}_{\chi}(A=a') \\
&= 0.6 \times (0.39285714285714285 * 0.875 + 0.25 * 0.125) \\
&+ 0.4 \times (0.8928571428571429 * 0.875 + 0.7500000000000001 * 0.125) \\
&= 0.575
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}_{\chi}(Y=1|do(A=0)) &= \sum_{m\in\{0,1\}} \mathbb{P}_{\chi}(M=m|A=0) \sum_{a'\in\{0,1\}} \mathbb{P}_{\chi}(Y=1|M=m,A=a')\mathbb{P}_{\chi}(A=a') \\
&= \mathbb{P}_{\chi}(M=1|A=0) \sum_{a'\in\{0,1\}} P(Y=1|M=1,A=a')\mathbb{P}_{\chi}(A=a') \\
&+ \mathbb{P}_{\chi}(M=0|A=0) \sum_{a'\in\{0,1\}} P(Y=1|M=0,A=a')\mathbb{P}_{\chi}(A=a') \\
&= 0.5 \times (0.6071428571428571 * 0.875 + 0.75 * 0.125 \\
&+ 0.5 \times (0.10714285714285714 * 0.875 + 0.25 * 0.125) \\
&= 0.375
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}_{\chi}(Y=0|do(A=0)) &= \sum_{m\in\{0,1\}} \mathbb{P}_{\chi}(M=m|A=0) \sum_{a'\in\{0,1\}} \mathbb{P}_{\chi}(Y=0|M=m,A=a')\mathbb{P}_{\chi}(A=a') \\
&= \mathbb{P}_{\chi}(M=1|A=0) \sum_{a'\in\{0,1\}} P(Y=0|M=1,A=a')\mathbb{P}_{\chi}(A=a') \\
&+ \mathbb{P}_{\chi}(M=0|A=0) \sum_{a'\in\{0,1\}} P(Y=0|M=0,A=a')\mathbb{P}_{\chi}(A=a') \\
&= 0.5 \times (0.39285714285714285 * 0.875 + 0.25 * 0.125) \\
&+ 0.5 \times (0.8928571428571429 * 0.875 + 0.7500000000000001 * 0.125) \\
&= 0.625
\end{aligned}
$$

2. Moral separation implies d-separation

(a) Show that all nodes in γ are in $V = \overline{\mathbf{an}_{\mathcal{G}}}(A \cup B \cup S)$.

Consider a node $\gamma_m \in \gamma$ where γ is a d-connecting path from A and B in $\mathcal{G}$ in the situation of given S. γ_m can unblock the path as a collider, where $\gamma_m \in S$ or one of its descendent is in S: $\text{de}_{\mathcal{G}}(\gamma_m) \in S$. When $\gamma_m \in S$, $\gamma_m \in V$ as $S \in V$. And when $\text{de}_{\mathcal{G}}(\gamma_m) \in S$, $\gamma_m \in \text{an}_{\mathcal{G}}(S)$ thus $\gamma_m \in V$. Thus, for both of the cases, $\gamma_m \in V$ as V contains the set S and their ancestors.

When γ_m is a non-collider, then it should not be conditioned so as not to block the path and construct d-connecting path γ, thus $\gamma_m \notin S$. As a non-collider, it can either be a confounder (e.g., $A \leftarrow \gamma_m \rightarrow B$) or in the path between A and B (e.g., $A \rightarrow \gamma_m \rightarrow B$). If it is a confounder in a path, $\gamma_m \in \text{an}_{\mathcal{G}}(A)$ and $\gamma_m \in \text{an}_{\mathcal{G}}(B)$ thus $\gamma_m \in V$. If γ_m is in the middle of the path, without any node in between γ_m and B that are not conditioned then $\gamma_m \in \text{an}_{\mathcal{G}}(B) \in V$. If there is any node γ_n in between γ_m and B that are conditioned ($\gamma_n \in S$), then it should be a collider to make γ a d-connecting path, which makes $\gamma_m \in \text{an}_{\mathcal{G}}(S) \in V$.

(b) Show that there is a path in $\overline{\mathcal{G}[V]}$ from A to B which does not pass through S.

Let $\gamma_m \in S$ be a node in $\mathcal{G}[V]$, then it should be a collider to make the path that it has contained a d-connecting path. Then in the moral graph $\overline{\mathcal{G}[V]}$ its parents are connected, proving the existence of the path from A to B does not pass S.

3. Simpson's paradox
(a) What is $\mathbb{P}_\chi(Y=1|A=N)$? What is $\mathbb{P}_\chi(Y=1|A=O)$?

$$\begin{aligned}\mathbb{P}_\chi(Y=1|A=N) &= \mathbb{P}_\chi(Y=1|S=L,A=N)\mathbb{P}_\chi(S=L)+\mathbb{P}_\chi(Y=1|S=R,A=N)\mathbb{P}_\chi(S=R)\\ &= 0.73\times 0.49+0.93\times 0.51\\ &= 0.832\end{aligned}$$

$$\begin{aligned}\mathbb{P}_\chi(Y=1|A=O) &= \mathbb{P}_\chi(Y=1|S=L,A=O)\mathbb{P}_\chi(S=L)+\mathbb{P}_\chi(Y=1|S=R,A=O)\mathbb{P}_\chi(S=R)\\ &= 0.69\times 0.49+0.87\times 0.51\\ &= 0.7818\end{aligned}$$

(b) What is $\mathbb{P}_\chi(Y=1|do(A=N))$? What is $\mathbb{P}_\chi(Y=1|do(A=O))$?
As the given graph in Figure 2.1(b) satisfies the backdoor criterion, we can calculate the interventional distribution by summing up all the backdoor set as below, where we get exactly the same value from (a).

$$\begin{aligned}\mathbb{P}_\chi(Y=1|do(A=N)) &= \sum_{s\in L,R}\mathbb{P}_\chi(Y=1|S=s,A=N)\mathbb{P}_\chi(S=s)\\ &= \mathbb{P}_\chi(Y=1|S=L,A=N)\mathbb{P}_\chi(S=L)+\mathbb{P}_\chi(Y=1|S=R,A=N)\mathbb{P}_\chi(S=R)\\ &= 0.73\times 0.49+0.93\times 0.51\\ &= 0.832\end{aligned}$$

$$\begin{aligned}\mathbb{P}_\chi(Y=1|do(A=O)) &= \sum_{s\in L,R}\mathbb{P}_\chi(Y=1|S=s,A=O)\mathbb{P}_\chi(S=s)\\ &= \mathbb{P}_\chi(Y=1|S=L,A=O)\mathbb{P}_\chi(S=L)+\mathbb{P}_\chi(Y=1|S=R,A=O)\mathbb{P}_\chi(S=R)\\ &= 0.69\times 0.49+0.87\times 0.51\\ &= 0.7818\end{aligned}$$

4. Instrumental Variables
The value of $\hat{\beta}_{aw}$ and $\hat{\beta}_{yw}$ were estimated by the coefficient of the linear regression model of A onto W and Y onto W, respectively. The value of $\hat{\beta}_{aw}$, $\hat{\beta}_{yw}$, and the ratio $\frac{\hat{\beta}_y w}{\hat{\beta}_a w}$ is reported in Table 1 below. The ratio $\frac{\hat{\beta}_y w}{\hat{\beta}_a w}$ is estimated similar to the coefficient $\beta_{ay}=7.5$ from the given equation. All the estimated values were rounded to second decimal places. The code used to calculate this is in Figure 5.

Table 1: Regression coefficients of instrumental variable models.

$\beta_w a$	$\hat{\beta}_a w$	$\hat{\beta}_y w$	$\frac{\hat{\beta}_y w}{\hat{\beta}_a w}$
0.05	0.33	2.49	7.63
0.5	0.73	5.54	7.60
5	4.53	33.85	7.48

A Appendix

```
import  numpy as np

class Bernoulli:
    def __init__(self, p):
        self.p = p
    def probability(self, x=1):
        if x == 1:
            return self.p
        elif x == 0:
            return 1-self.p
```

```
U = {0: 0.5, 1: 0.5}
PA_U = lambda x, y: Bernoulli(x/4).probability(y)
PM_A = lambda x, y: Bernoulli(0.5+0.1*x).probability(y)
PY_MA = lambda x, y, z: Bernoulli(x/2+y/4).probability(z)
```

```
PYAMU = {}
for u in [0,1]:
    for a in [0,1]:
        for m in [0,1]:
            for y in [0,1]:
                PYAMU[(y,a,m,u)] = U[u] * PA_U(u,a) * PM_A(a,m) * PY_MA(m, u, y)
```

```
PYMA = 0 #P(Y=1, M=0, A=0)
PYMA1 = 0 #P(Y=1, M=0, A=1)
PMA0 = 0 #P(M=0, A=0)
PMA1 = 0 #P(M=0, A=1)
for u in [0,1]:
    PYMA += PYAMU[(1, 0, 0, u)]
    PYMA1 += PYAMU[(1, 1, 0, u)]
    for y in [0,1]:
        PMA0 += PYAMU[(y, 0, 0, u)]
        PMA1 += PYAMU[(y, 1, 0, u)]
print("P(Y=1|M=0, A=0 = {}, P(Y=1|M=0, A=1) = {}".format(PYMA/PMA0, PYMA1/PMA1))
print(PYMA, PYMA1, PMA0, PMA1)
```

```
P(Y=1|M=0, A=0 = 0.10714285714285714, P(Y=1|M=0, A=1) = 0.25
0.046875 0.0125 0.4375 0.05
```

Figure 1: Probability Calculation for question 1b.

Interventional

```
U = {0: 0.5, 1: 0.5}
M_doA1 = {0: 0.4, 1: 0.6}
PY_MA = lambda x, y, z: Bernoulli(x/2+y/4).probability(z)

PYMU = {}
for u in [0,1]:
    for m in [0,1]:
        for y in [0,1]:
            PYMU[(y,m,u)] = U[u] * M_doA1[m] * PY_MA(m, u, y)
```

```
PY_doA1 = 0 #P(Y=1)
for u in [0,1]:
    for m in [0,1]:
        PY_doA1+=PYMU[(1, m, u)]
print(PY_doA1)
```

```
0.425
```

```
U = {0: 0.5, 1: 0.5}
M_doA0 = {0: 0.5, 1: 0.5}
PY_MA = lambda x, y, z: Bernoulli(x/2+y/4).probability(z)

PYMU0 = {}
for u in [0,1]:
    for m in [0,1]:
        for y in [0,1]:
            PYMU0[(y,m,u)] = U[u] * M_doA0[m] * PY_MA(m, u, y)
```

```
PY_doA1 = 0 #P(Y=1)
for u in [0,1]:
    for m in [0,1]:
        PY_doA1+=PYMU0[(0, m, u)]
print(PY_doA1)
```

```
0.625
```

Figure 2: Probability Calculation for question 1d.

Backdoor Adjustment

```python
PYUA1 = 0 #P(Y=1, U=1, A=1)
PYUA0 = 0 #P(Y=1, U=0, A=1)
PUA1 = 0 #P(U=1, A=1)
PUA0 = 0 #P(U=0, A=1)

for m in [0,1]:
    PYUA1 += PYAMU[(1, 1, m, 1)]
    PYUA0 += PYAMU[(1, 1, m, 0)]
    for y in [0,1]:
        PUA1 += PYAMU[(y, 1, m, 1)]
        PUA0 += PYAMU[(y, 1, m, 0)]

print(PYUA1, PYUA0, PUA1, PUA0)
```

```
0.06874999999999999 0.0 0.125 0.0
```

```python
PYUA1 = 0 #P(Y=0, U=1, A=1)
PYUA0 = 0 #P(Y=0, U=0, A=1)
PUA1 = 0 #P(U=1, A=1)
PUA0 = 0 #P(U=0, A=1)

for m in [0,1]:
    PYUA1 += PYAMU[(0, 1, m, 1)]
    PYUA0 += PYAMU[(0, 1, m, 0)]
    for y in [0,1]:
        PUA1 += PYAMU[(y, 1, m, 1)]
        PUA0 += PYAMU[(y, 1, m, 0)]

print(PYUA1, PYUA0, PUA1, PUA0)
```

```
0.05625000000000001 0.0 0.125 0.0
```

```python
PYUA1 = 0 #P(Y=1, U=1, A=0)
PYUA0 = 0 #P(Y=1, U=0, A=0)
PUA1 = 0 #P(U=1, A=1)
PUA0 = 0 #P(U=0, A=1)

for m in [0,1]:
    PYUA1 += PYAMU[(1, 0, m, 1)]
    PYUA0 += PYAMU[(1, 0, m, 0)]
    for y in [0,1]:
        PUA1 += PYAMU[(y, 0, m, 1)]
        PUA0 += PYAMU[(y, 0, m, 0)]

print(PYUA1, PYUA0, PUA1, PUA0)
print("P(Y=1|U=1, A=1) = {}, P(Y=1|U=0, A=1) = {}".format(PYUA1/PUA1, PYUA0/PUA0))
print(PYUA1/PUA1*0.5+PYUA0/PUA0*0.5)
```

```
0.1875 0.125 0.375 0.5
P(Y=1|U=1, A=1) = 0.5, P(Y=1|U=0, A=1) = 0.25
0.375
```

```python
PYUA1 = 0 #P(Y=0, U=1, A=0)
PYUA0 = 0 #P(Y=0, U=0, A=0)
PUA1 = 0 #P(U=1, A=1)
PUA0 = 0 #P(U=0, A=1)

for m in [0,1]:
    PYUA1 += PYAMU[(0, 0, m, 1)]
    PYUA0 += PYAMU[(0, 0, m, 0)]
    for y in [0,1]:
        PUA1 += PYAMU[(y, 0, m, 1)]
        PUA0 += PYAMU[(y, 0, m, 0)]

print(PYUA1, PYUA0, PUA1, PUA0)
print("P(Y=0|U=1, A=1) = {}, P(Y=0|U=0, A=1) = {}".format(PYUA1/PUA1, PYUA0/PUA0))
print(PYUA1/PUA1*0.5+PYUA0/PUA0*0.5)
```

```
0.1875 0.375 0.375 0.5
P(Y=0|U=1, A=1) = 0.5, P(Y=0|U=0, A=1) = 0.75
0.625
```

Figure 3: Probability Calculation for question 1e.

Frontdoor Adjustment

```python
PYMA = 0 #P(Y=1, M=0, A=0)
PYMA1 = 0 #P(Y=1, M=0, A=1)
PYM1A = 0 #P(Y=1, M=1, A=0)
PYM1A1 = 0 #P(Y=1, M=1, A=1)
PM0A = 0 #P(M=0, A=0)
PM0A1 = 0 #P(M=0, A=1)
PM1A = 0 #P(M=1, A=0)
PM1A1 = 0 #P(M=1, A=1)
PA = 0 #P(A=0)

for u in [0,1]:
    PYMA += PYAMU[(1, 0, 0, u)]
    PYM1A += PYAMU[(1, 0, 1, u)]
    PYMA1 += PYAMU[(1, 1, 0, u)]
    PYM1A1 += PYAMU[(1, 1, 1, u)]
    for y in [0,1]:
        PM0A += PYAMU[(y, 0, 0, u)]
        PM1A += PYAMU[(y, 0, 1, u)]
        PM0A1 += PYAMU[(y, 1, 0, u)]
        PM1A1 += PYAMU[(y, 1, 1, u)]
        for m in [0,1]:
            PA += PYAMU[(y, 0, m, u)]
print("P(Y=1|M=1, A=0) = {}, P(Y=1|M=0, A=0) = {}".format(PYM1A/PM1A, PYMA/PM0A))
print("P(Y=1|M=1, A=1) = {}, P(Y=1|M=0, A=1) = {}".format(PYM1A1/PM1A1, PYMA1/PM0A1))
print("P(M=1| A=0) = {}, P(M=0|A=0) = {}".format(PM1A/PA, PM0A/PA))
print("P(M=1| A=1) = {}, P(M=0|A=1) = {}".format(PM1A1/(1-PA), PM0A1/(1-PA)))
print(PYMA, PYM1A, PM0A, PM1A ,PA)

print(0.6 * (0.6071428571428571*0.875+0.75*0.125)+0.4 * (0.10714285714285714*0.875+0.25*0.125))
print(0.5 * (0.6071428571428571*0.875+0.75*0.125)+0.5 * (0.10714285714285714*0.875+0.25*0.125))
```

```
P(Y=1|M=1, A=0) = 0.6071428571428571, P(Y=1|M=0, A=0) = 0.10714285714285714
P(Y=1|M=1, A=1) = 0.75, P(Y=1|M=0, A=1) = 0.25
P(M=1| A=0) = 0.5, P(M=0|A=0) = 0.5
P(M=1| A=1) = 0.6, P(M=0|A=1) = 0.4
0.046875 0.265625 0.4375 0.4375 0.875
0.425
0.375
```

```python
PYMA = 0 #P(Y=0, M=0, A=0)
PYMA1 = 0 #P(Y=0, M=0, A=1)
PYM1A = 0 #P(Y=0, M=1, A=0)
PYM1A1 = 0 #P(Y=0, M=1, A=1)
PM0A = 0 #P(M=0, A=0)
PM0A1 = 0 #P(M=0, A=1)
PM1A = 0 #P(M=1, A=0)
PM1A1 = 0 #P(M=1, A=1)
PA = 0 #P(A=0)

for u in [0,1]:
    PYMA += PYAMU[(0, 0, 0, u)]
    PYM1A += PYAMU[(0, 0, 1, u)]
    PYMA1 += PYAMU[(0, 1, 0, u)]
    PYM1A1 += PYAMU[(0, 1, 1, u)]
    for y in [0,1]:
        PM0A += PYAMU[(y, 0, 0, u)]
        PM1A += PYAMU[(y, 0, 1, u)]
        PM0A1 += PYAMU[(y, 1, 0, u)]
        PM1A1 += PYAMU[(y, 1, 1, u)]
        for m in [0,1]:
            PA += PYAMU[(y, 0, m, u)]
print("P(Y=0|M=1, A=0) = {}, P(Y=0|M=0, A=0) = {}".format(PYM1A/PM1A, PYMA/PM0A))
print("P(Y=0|M=1, A=1) = {}, P(Y=0|M=0, A=1) = {}".format(PYM1A1/PM1A1, PYMA1/PM0A1))
print("P(M=1| A=0) = {}, P(M=0|A=0) = {}".format(PM1A/PA, PM0A/PA))
print("P(M=1| A=1) = {}, P(M=0|A=1) = {}".format(PM1A1/(1-PA), PM0A1/(1-PA)))
print(PYMA, PYM1A, PM0A, PM1A ,PA)

print(0.5*(0.39285714285714285*0.875+0.25*0.125)+ 0.5*(0.8928571428571429*0.875+0.7500000000000001*0.125))
print(0.6*(0.39285714285714285*0.875+0.25*0.125)+ 0.4*(0.8928571428571429*0.875+0.7500000000000001*0.125))
```

```
P(Y=0|M=1, A=0) = 0.39285714285714285, P(Y=0|M=0, A=0) = 0.8928571428571429
P(Y=0|M=1, A=1) = 0.25, P(Y=0|M=0, A=1) = 0.7500000000000001
P(M=1| A=0) = 0.5, P(M=0|A=0) = 0.5
P(M=1| A=1) = 0.6, P(M=0|A=1) = 0.4
0.390625 0.171875 0.4375 0.4375 0.875
0.625
0.575
```

Figure 4: Probability Calculation for question 1f.

Calculating $\hat{\beta}_{aw}$

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

samples_005 = pd.read_csv('./samples_0.05.csv', delimiter=' ', names=['W', 'A', 'Y'])
samples_05 = pd.read_csv('./samples_0.5.csv', delimiter=' ', names=['W', 'A', 'Y'])
samples_5 = pd.read_csv('./samples_5.csv', delimiter=' ', names=['W', 'A', 'Y'])

# 0.05
a_005 = np.asarray(samples_005['A']).reshape(-1, 1)
w_005 = np.asarray(samples_005['W']).reshape(-1, 1)

reg_005 = LinearRegression().fit(w_005, a_005)
beta_aw_005 = reg_005.coef_
print (reg_005.score(w_005, a_005), beta_aw_005)
```

```
0.0009904562115915505 [[0.32646744]]
```

```
# 0.5
a_05 = np.asarray(samples_05['A']).reshape(-1, 1)
w_05 = np.asarray(samples_05['W']).reshape(-1, 1)

reg_05 = LinearRegression().fit(w_05, a_05)
beta_aw_05 = reg_05.coef_
print (reg_05.score(w_05, a_05), beta_aw_05)
```

```
0.004832409817257077 [[0.72908667]]
```

```
# 5
a_5 = np.asarray(samples_5['A']).reshape(-1, 1)
w_5 = np.asarray(samples_5['W']).reshape(-1, 1)

reg_5 = LinearRegression().fit(w_5, a_5)
beta_aw_5 = reg_5.coef_
print (reg_5.score(w_5, a_5), beta_aw_5)
```

```
0.1619688187872389 [[4.52729139]]
```

Calculating $\hat{\beta}_{yw}$

```
# 0.05
y_005 = np.asarray(samples_005['Y']).reshape(-1, 1)

reg_005 = LinearRegression().fit(w_005, y_005)
beta_yw_005 = reg_005.coef_
print (reg_005.score(w_005, y_005), beta_yw_005)
```

```
0.0009603810161366022 [[2.49075696]]
```

```
# 0.5
y_05 = np.asarray(samples_05['Y']).reshape(-1, 1)

reg_05 = LinearRegression().fit(w_05, y_05)
beta_yw_05 = reg_05.coef_
print (reg_05.score(w_05, y_05), beta_yw_05)
```

```
0.004650886498636875 [[5.54277931]]
```

```
# 5
y_5 = np.asarray(samples_5['Y']).reshape(-1, 1)

reg_5 = LinearRegression().fit(w_5, y_5)
beta_yw_5 = reg_5.coef_
print (reg_5.score(w_5, y_5), )
```

```
0.15248865899664366 [[33.84799159]]
```

```
print(beta_yw_005/beta_aw_005)
print(beta_yw_05/beta_aw_05)
print(beta_yw_5/beta_aw_5)
```

```
[[7.62941922]]
[[7.60235993]]
[[7.47643319]]
```

Figure 5: Linear Regression Model Code for question 4.