Causal Inference - HW2

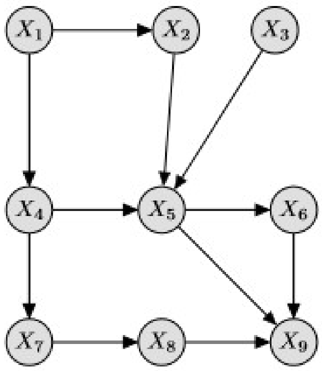
Maxime TCHIBOZO (mt3390)

In [1]:
```
import numpy as np
import pandas as pd
```

Exercise 1 - Find a set Z that d-separates two nodes

In [6]:
```
from IPython.display import Image
path = 'C:/Users/Max Tchibozo/Desktop/CAUSAL INFERENCE/HW2/'
Image(path+'Ex1graph.png')
```

Out[6]:



(a) find Z that d-separates X_1 from X_9

The acyclic paths from X_1 to X_9 are:
$p_1 = (X_1, X_2, X_5, X_4, X_7, X_8, X_9, p_2 = (X_1, X_2, X_5, X_6, X_9), p_3 = (X_1, X_2, X_5, X_9), p_4 = (X_1, X_4, X_5, X_6, .$
$= (X_1, X_4, X_7, X_8, X_9)$

X_2 is in the middle of a chain so it blocks (p_1, p_2, p_3), and X_4 is in the middle of a chain so it blocks (p_4, p_5, p_6).

The set $Z = \{X_2, X_4\}$ d-separates X_1 from X_9.

(b) find Z that d-separates X_4 from X_6

The acyclic paths from x_4 to X_6 are :
$p_1 = (X_4, X_1, X_2, X_5, X_6), p_2 = (X_4, X_1, X_2, X_5, X_9, X_6), p_3 = (X_4, X_5, X_6), p4 = (X_4, X_5, X_9, X_6), p_5 = ($
$= (X_4, X_7, X_8, X_9, X_6)$

X_5 is the middle of a chain for (p_1, p_2, p_3, p_4). (p_5, p_6) are naturally blocked by the collider in X_9.

X_5 is a collider, so adding it to Z will unblock paths containing both tails of the collider. Thankfully, none of the paths from X_4 to X_6 go through the collider and its tails, so no path is affected. The set $Z = \{X_5\}$ d-separates X_4 from X_6.

(c) find Z that d-separates X_5 from X_8

The acyclic paths from X_5 to X_8 are :
$p_1 = (X_5, X_2, X_1, X_4, X_7, X_8), p_2 = (X_5, X_4, X_7, X_8), p_3 = (X_5, X_6, X_9, X_8), p_4 = (X_5, X_9, X_8)$.

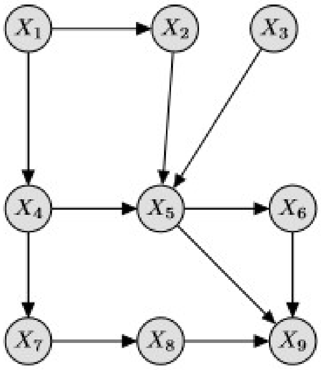
p_3 and p_4 contain a collider in X_9. So long as we do not include X_9, any Z set will d-separate p_3 and p_4. X_4 is in the middle of a chain for p_1 and fork for p_2.

The set $Z = \{X_4\}$ d-separates X_5 from X_8.

Exercise 2 : Find a backdoor set for two nodes

In [7]:
```
Image(path+'Ex1graph.png')
```

Out[7]:



(a) What is a back-door set for (X_1, X_9)?

All nodes except for X_3 are descendants of X_1. There is no arrow into X_1. Therefore, $Z = \{X_3\}$ is a back-door set for X_1 on X_9.

(b) What is a back-door set for (X_9, X_1)?

X_9 has no descendants. All paths between X_9 and X_1 have arrows into X_9. Blocking X_2 and X_4 blocks all of these possible paths. Therefore, $Z = \{X_2, X_4\}$ is a back-door set for X_9 and X_1.

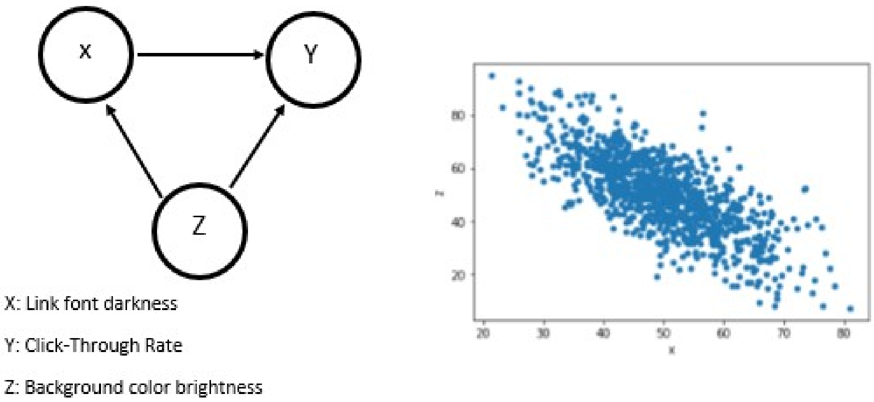
(c) What is a back-door set for (X_2, X_9)?

X_5, X_6,and X_9 are descendants of X_2. We must avoid these nodes. All paths with an arrow into X_2 go through X_1. Therefore, $Z = \{X_1\}$ is a back-door set for X_2 and X_9.

Exercise 3 : Inferring policy outcomes

In [13]:
```
Image(path+'Ex3graph.png')
```

Out[13]:



First, we assume that the causal graph of the problem is as shown above. In this particular instance, the background color brightness Z is a confounder for X and Y.

According to the above causal graph $\{Z\}$ is a valid backdoor set : It is not a descendent of X, and it blocks the path of X to Y with an arrow into X. To measure the effect of X on Y, we can condition on Z.

(a) Effect of setting X=50 when Z=50

The quantity we are trying to estimate is $\mathbb{E}[Y|do(X=50, Z=50)]$. Since there are many data points in this region, the backdoor adjustment formula is valid, and we can reasonably condition on X and Z.

To estimate the effect of setting X to 50 and Z to 50, we should compute the empirical mean of all Click-Through Rates when X=50 and Z=50 in our historical data. Judging by the density of the $P(X, Z)$ plot in the (X=50,Z=50) region, there should be several click-through rate points, and our empirical mean will not be too biased.

(b) Effect of setting X=30, independently of Z

In this instance, the quantity we are trying to evaluate is $\mathbb{E}[Y|do(X=30)]$.

Here, we are faced with a problem : Z is a discrete variable with many values (high cardinality). In our data, we do not have many observations of $(X=30, Z)$ points. This means that for X=30, there are strata of Z that do not contain any data points. The solution here is to drop the strata of Z which have no data points for X=30. This will undoubtedly introduce bias in our estimator of $\mathbb{E}[Y|do(X=30)]$, but it will still probably provide a better estimator than the naïve estimator.

Let Z^* be the set of Z restricted to existing data points

We can stratify on the values of Z^* using the Robins G-Formula applied to conditional expectations : $\mathbb{E}[Y|do(X=30)] = \sum_z \mathbb{E}[Y|do(X=30), Z^*]p(Z^*)$ to estimate the policy outcome.

(c) Effect of setting X=10, independently of Z

In this instance, the quantity we are trying to evaluate is $\mathbb{E}[Y|do(X=10)]$. Unfortunately, we are clearly outside of the support of the data, so we can not drop observations of (X, Z) nor extrapolate $P(X, Z)$ in this region.

We can no longer use the previous argument, as we have no data points for $Z(X=10)$. To assume a distribution for P(X=30,Z) would be to guess completely. In this example, we can not infer the outcome of setting these parameters.

Exercise 4

In [48]:
```
N=2000

u1=np.random.normal(size=N)
u2=np.random.binomial(1,p=1/(1+np.exp(-u1)),size=N)
x=np.random.binomial(1,p=1/(1+np.exp(-u1)))
y=np.random.normal(x+u2)
df = pd.DataFrame({'U1':u1,'U2':u2,'X':x,'Y':y})
df.head()
```

Out[48]:

	U1	U2	X	Y
0	1.354964	0	0	-0.187165
1	-0.605305	1	0	2.007762
2	0.444342	1	0	1.317971
3	-0.583783	0	1	-0.004516
4	0.696892	1	1	3.470735

(a) Causal graph for the system and average treatment effect.

To measure the average treatement effect of X on Y, let us first define the data generating process from a functional point of view.

$U1 = \epsilon_1$ where $\epsilon_1 \sim \mathcal{N}(0,1)$

$U2 = \epsilon_2$ where $\epsilon_2 \sim \mathcal{B}(\frac{1}{1+exp(-U1)})$

$X = \epsilon_X$ where $\epsilon_X \sim \mathcal{B}(\frac{1}{1+exp(-U1)})$

$Y = X + U2 + \epsilon_Y$ where $\epsilon_Y \sim \mathcal{N}(0,1)$

$\delta = \mathbb{E}[Y^{(X=1)}] - \mathbb{E}[Y^{(X=0)}]$

$= \mathbb{E}[1 + (U2 + \epsilon_Y)^{(X=1)}] - \mathbb{E}[0 + (U2 + \epsilon_Y)^{(X=0)}]$

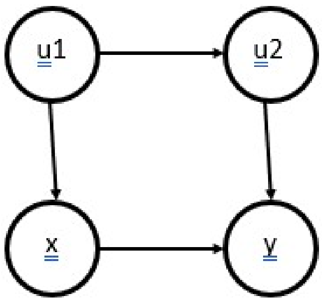
$= 1$ as U_2 and ϵ_Y are independent of X

The average treatment effect of X on Y is $\delta = 1$

The causal graph of the system is the following.

In [10]:
```
#Causal Graph
Image(path+'Ex4graph.png')
```

Out[10]:



(b) What are at least two sets of variables that satisfy the back-door criterion for the effect of X on Y

$Z = \{U_2\}$ is not a descendant of X, and blocks the path between X and Y with an arrow into X (it is the middle of a chain). It satisfies the backdoor criterion for the effect of X on Y.

$Z = \{U_1\}$ is also not a descendant of X, and blocks the path between X and Y with an arrow into X (it is the middle of a fork). It satisfies the backdoor criterion for the effect of X on Y.

(c) Use stratification to estimate the average treatment effect of X on Y

Let us first consider the naïve estimate:

In [52]:
```
delta_naive = df.groupby("X").mean()["Y"][1]-df.groupby("X").mean()["Y"][0]
print("delta naive : "+str(delta_naive))
```

```
delta naive : 1.1757299005877941
```

Since $\{U_2\}$ is a satisfying backdoor set, we will condition on U_2 since it is a discrete variable (easier to define strata).

We will then measure δ with the following formula : $\delta = \sum_{U2}(\mathbb{E}[Y|X=1,U2] - E[Y|X=0,U2])P(U2)$

In [56]:
```
#U2-level outcomes in the X=1 and X=0 groups (E[Y|X] groups)
X_1 = df[df.X==1].groupby("U2").mean().reset_index()
X_0 = df[df.X==0].groupby("U2").mean().reset_index()
```

In [71]:
```
#We compute the weights
w_1 = (X_1[X_1.U2==1]["Y"] - X_0[X_0.U2==1]["Y"]).mean() #the .mean is only to c
onvert dataframe to float
w_0 = (X_1[X_1.U2==0]["Y"] - X_0[X_0.U2==0]["Y"]).mean()

#We compute the probability distribution of U2
p_1 = len(df[df.U2==1])/len(df)
p_0 = len(df[df.U2==0])/len(df)
```

In [73]:
```
delta_U2 = w_1*p_1+w_0*p_0
print("delta when conditioning on U2 : "+str(delta_U2))
```

```
delta when conditioning on U2 : 1.0032044491264864
```

Conditioning on U2 has greatly reduced the bias, and provides a much better estimator of the average treatment effect than the naïve estimator.

In [ ]: