Connecting technology to intelligence

Simpson's Paradox analyzed through Causal Reasoning

pyAgrum User day 15/06/2021

Santiago Cortijo, PhD, Head R&D santiago.cortijo@scalnyx.com

SCALNYX S.A.S. 5 Avenue Ingres, 75016 Paris, France
info@scalnyx.com - Tel : +33 6 45 32 83 70

www.scalnyx.com

This presentation is based on:

1. **Understanding the Simpson's Paradox**, by Judea Pearl (2013)

2. Causality Course, by Brady Neal:

   https://www.bradyneal.com/causal-inference-course

Correlation ≠ Causation

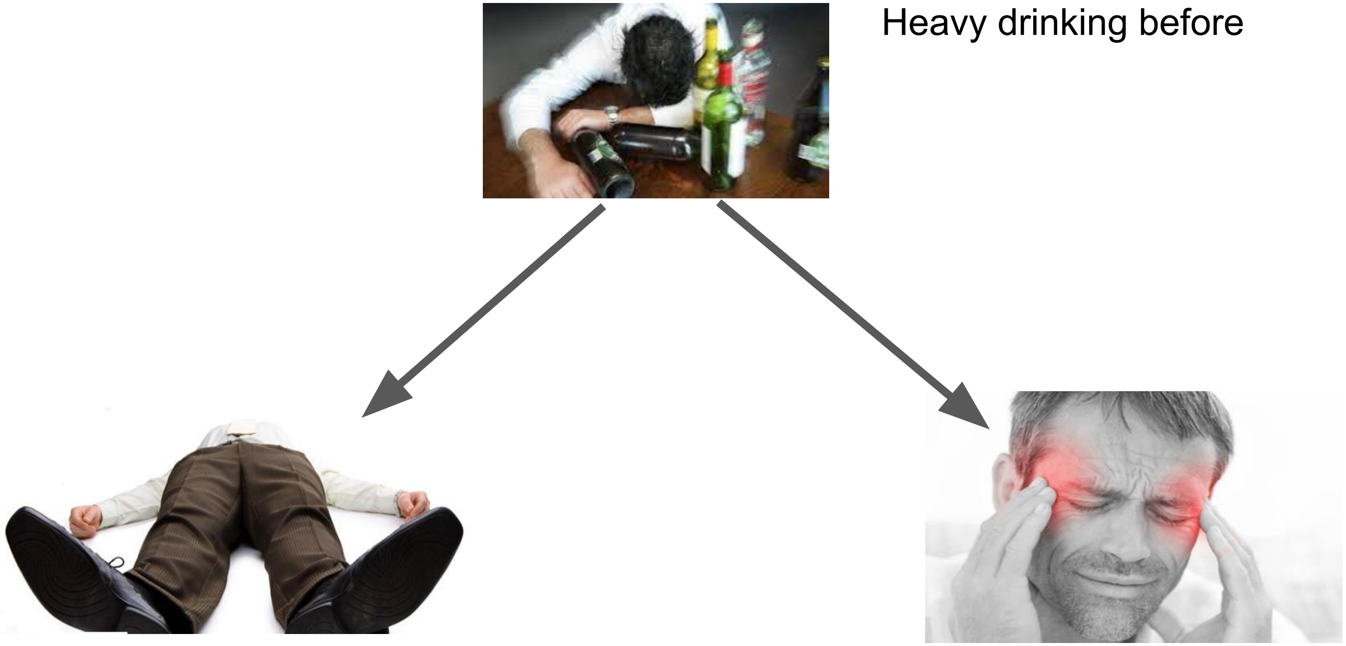
Motivating examples

Correlation ≠ Causation (1)

Sleep with shoes

Headache next morning.

Correlation ≠ Causation (2)

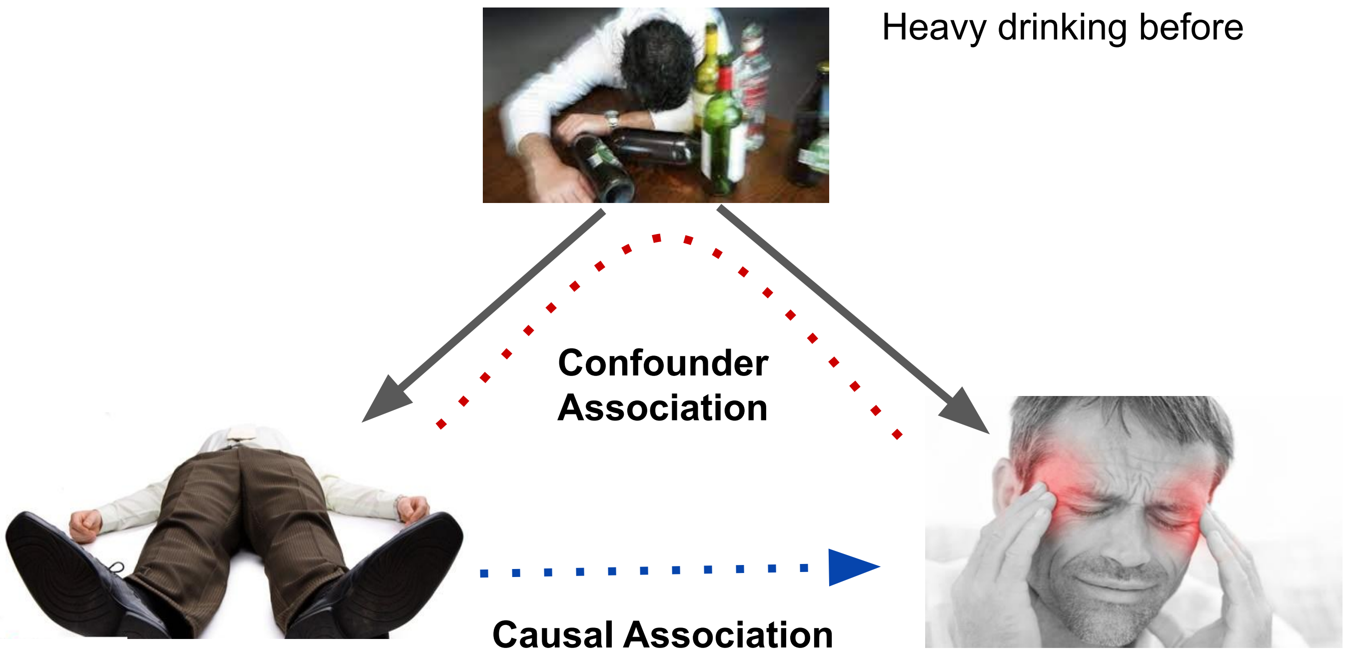


Heavy drinking before

Sleep with shoes

Headache next morning.

Correlation ≠ Causation (3)



Heavy drinking before

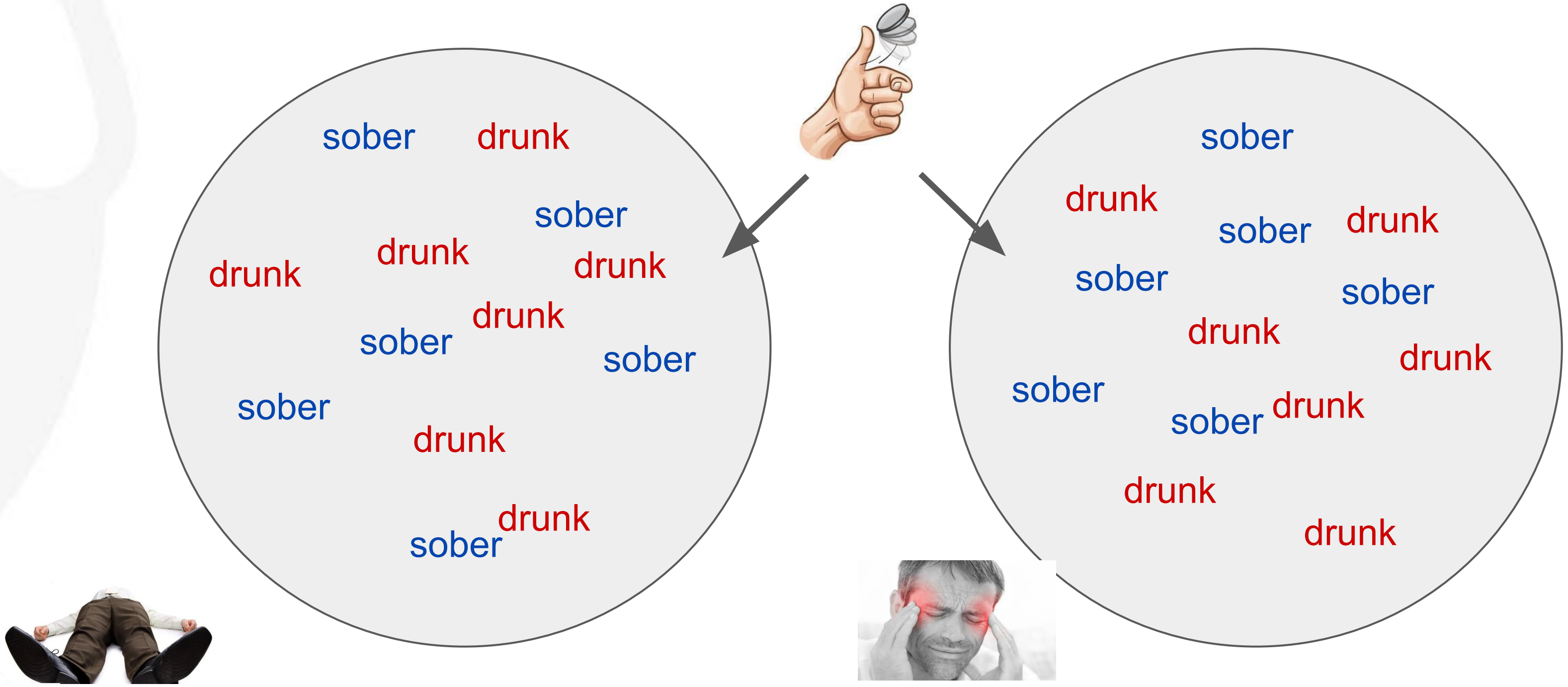
Confounder Association

Causal Association

Sleep with shoes

Headache next morning.

Randomized Control Trials (RCT), (1)



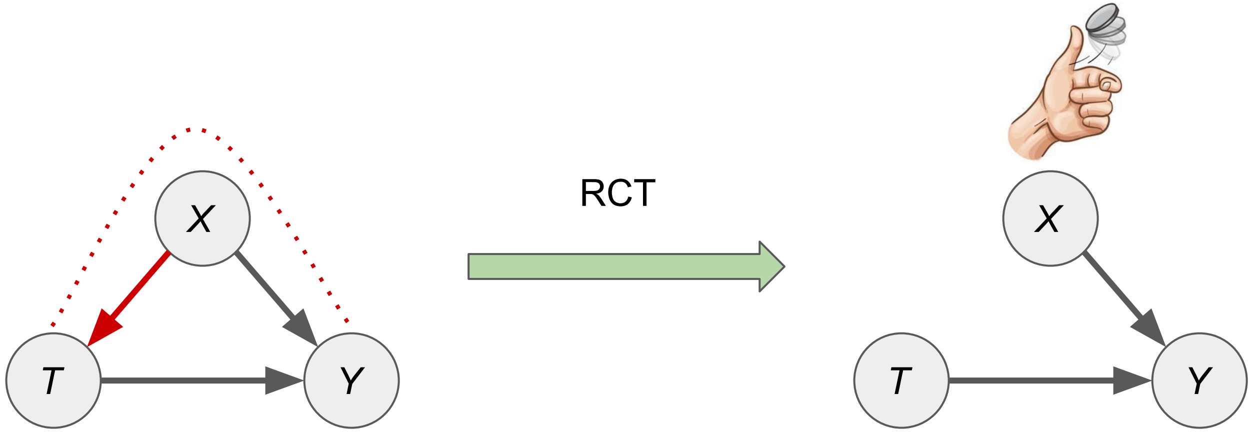
sober drunk
sober
drunk drunk
drunk
drunk
sober
sober
sober
drunk
drunk
sober

sober
drunk
sober drunk
sober
sober
drunk
drunk
sober
sober drunk
drunk
drunk

Slept with Shoes

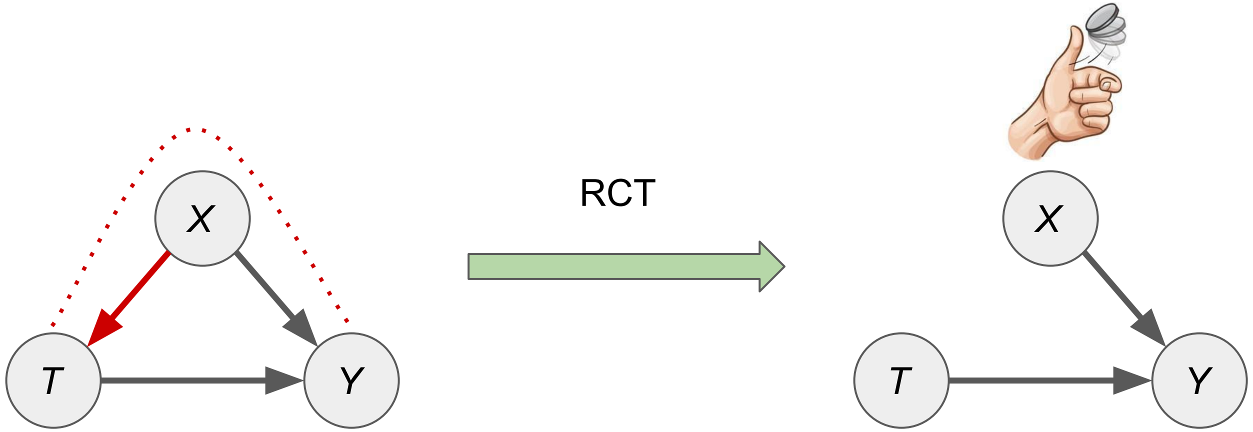
Slept Without Shoes

Randomized Control Trials (RCT), (2)



RCT eliminates confounding association

Randomized Control Trials (RCT), (3)



RCT eliminates confounding association

Sadly, RCT can are often expensive, unethical or even impossible.

Observational studies

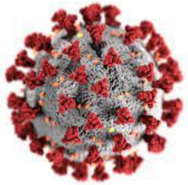
How do we measure causal effects

Motivating example : Treatment to choose (1)

T : Treatment (A or B)

☠️ Death rate table:

T	Total ☠️
A	**16 %** (240 / 1500)
B	19 % (105 / 550)



Decision Makers :
Need to invest in a Treatment. Which one? which is better, A or B ?

Motivating example : Treatment to choose (2)

T : Treatment (A or B)

☠️ Death rate table, with mode details about symptoms intensity:

T	Mild Symptoms.	Severe Symptoms	Total ☠️
A	15 % (210 / 1400)	30 % (30 / 100)	**16 %** (240 / 1500)
B	**10 %** (5 / 50)	**20 %** (100 / 500)	19 % (105 / 550)

Decision Makers:

Wait… what !?

Motivating example : Treatment to choose (3)

T : Treatment (A or B)

T	Mild	Severe	☠️
A	15 % (210 / 1400)	30 % (30 / 100)	**16 %** (240 / 1500)
B	**10 %** (5 / 50)	**20 %** (100 / 500)	19 % (105 / 550)

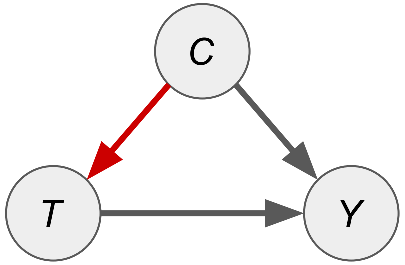
$$\underline{\frac{1400}{1500}}0.15 + \underline{\frac{100}{1500}}0.30 = 0.16$$

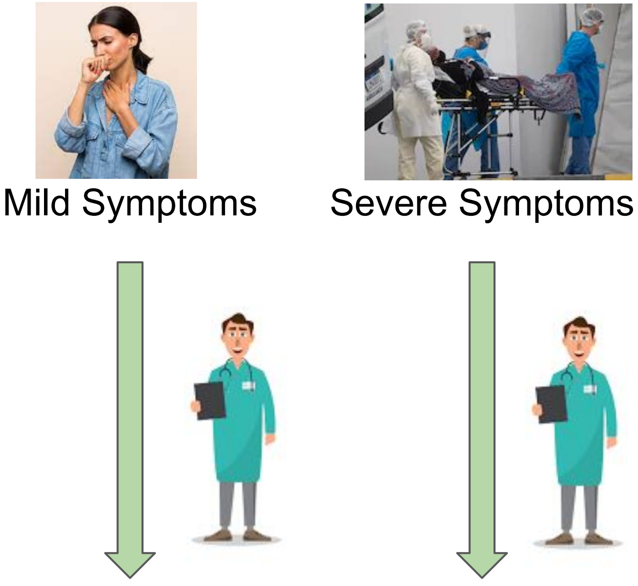
$$\underline{\frac{50}{550}}0.10 + \underline{\frac{500}{550}}0.20 = 0.19$$

Motivating example : When is B prefered ?

Causal Graph:

Symptom intensity (C) causes Treatment (T)





Mild Symptoms

Severe Symptoms

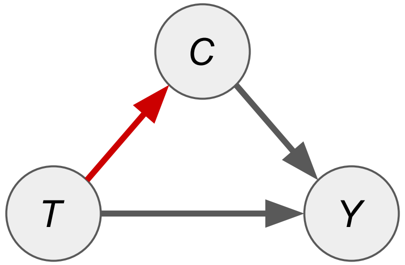
Treatment A

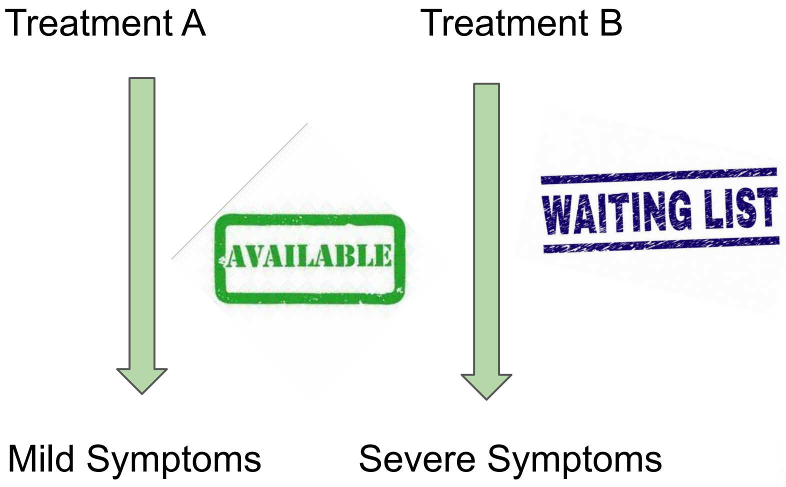
Treatment B

Motivating example : When is A prefered ?

Causal Graph:

Treatment (T) causes Symptom intensity (C) , due to waiting list





Treatment A

Treatment B

Mild Symptoms

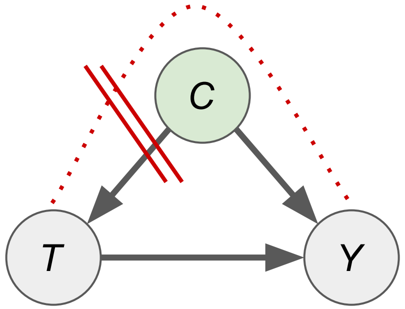
Severe Symptoms

Measure causal effects in observational data (1)

For our example, we **adjust/control for confounders**

T	Mild	Severe	☠️
A	15 % (210 / 1400)	30 % (30 / 100)	16 % (240 / 1500)
B	10 % (5 / 50)	20 % (100 / 500)	19 % (105 / 550)

$\mathrm{E}[Y|t, C=0]$ $\mathrm{E}[Y|t, C=1]$ $\mathrm{E}[Y|t]$



$$\mathrm{E}[Y \mid do(T=t)] = \mathrm{E}_C \mathrm{E}[Y \mid T=t]$$

Measure causal effects in observational data (2)

$$\mathrm{E}[Y \mid do(T=t)] = \mathrm{E}_C \mathrm{E}[Y \mid T=t]$$

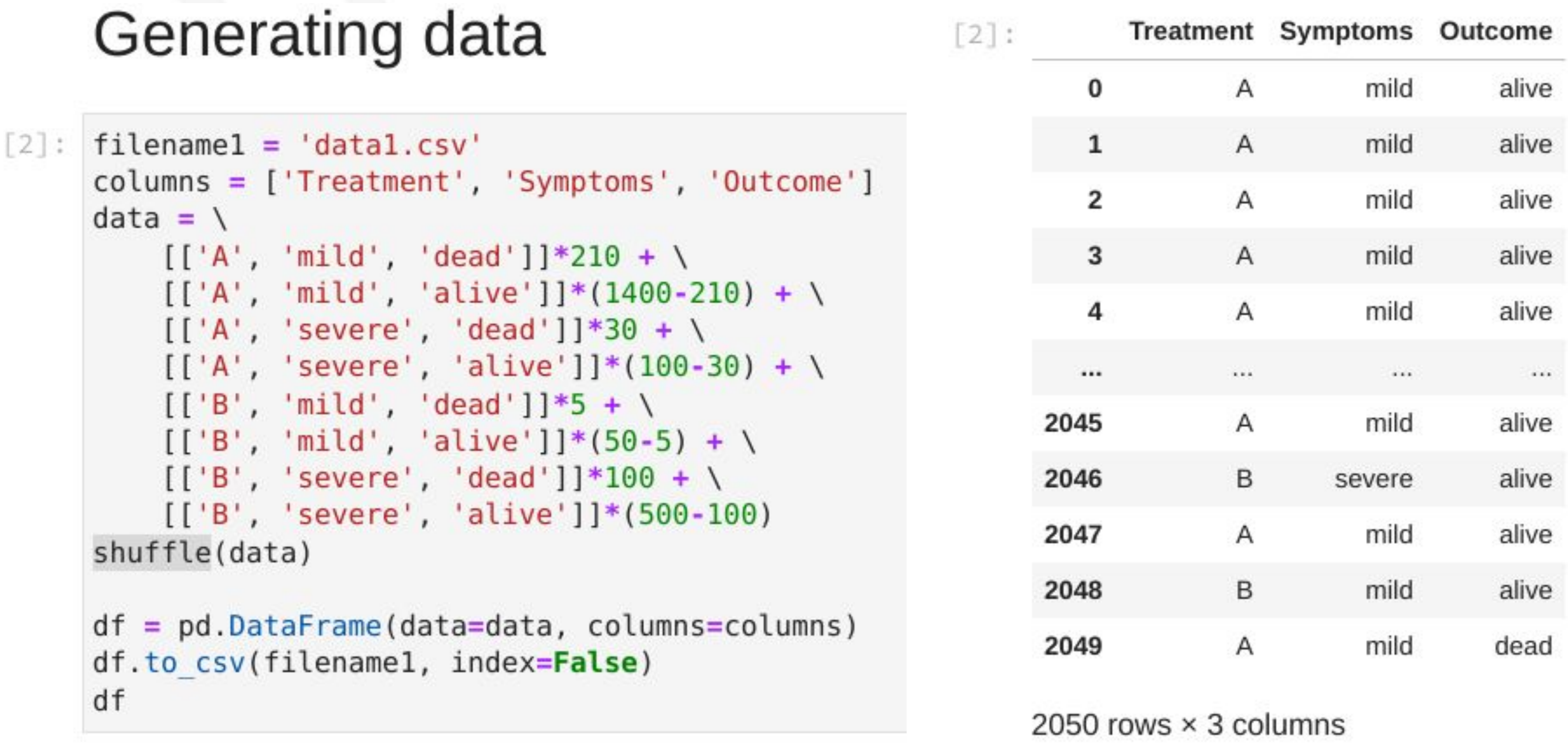
T	Mild	Severe	☠️	causal
A	15 % (210 / 1400)	30 % (30 / 100)	**16 %** (240 / 1500)	**19.4 %**
B	**10 %** (5 / 50)	**20 %** (100 / 500)	19 % (105 / 550)	12.9 %

$\mathrm{E}[Y|t, C=0]$ $\mathrm{E}[Y|t, C=1]$ $\mathrm{E}[Y|t]$ $\mathrm{E}[Y|do(t)]$

$$\frac{1450}{2050}0.15 + \frac{600}{2050}0.30 = 0.194$$

$$\frac{1450}{2050}0.10 + \frac{600}{2050}0.20 = 0.129$$

Done with pyAgrum (1)



Generating data

[2]:

```
filename1 = 'data1.csv'
columns = ['Treatment', 'Symptoms', 'Outcome']
data = \
    [['A', 'mild', 'dead']]*210 + \
    [['A', 'mild', 'alive']]*(1400-210) + \
    [['A', 'severe', 'dead']]*30 + \
    [['A', 'severe', 'alive']]*(100-30) + \
    [['B', 'mild', 'dead']]*5 + \
    [['B', 'mild', 'alive']]*(50-5) + \
    [['B', 'severe', 'dead']]*100 + \
    [['B', 'severe', 'alive']]*(500-100)
shuffle(data)

df = pd.DataFrame(data=data, columns=columns)
df.to_csv(filename1, index=False)
df
```

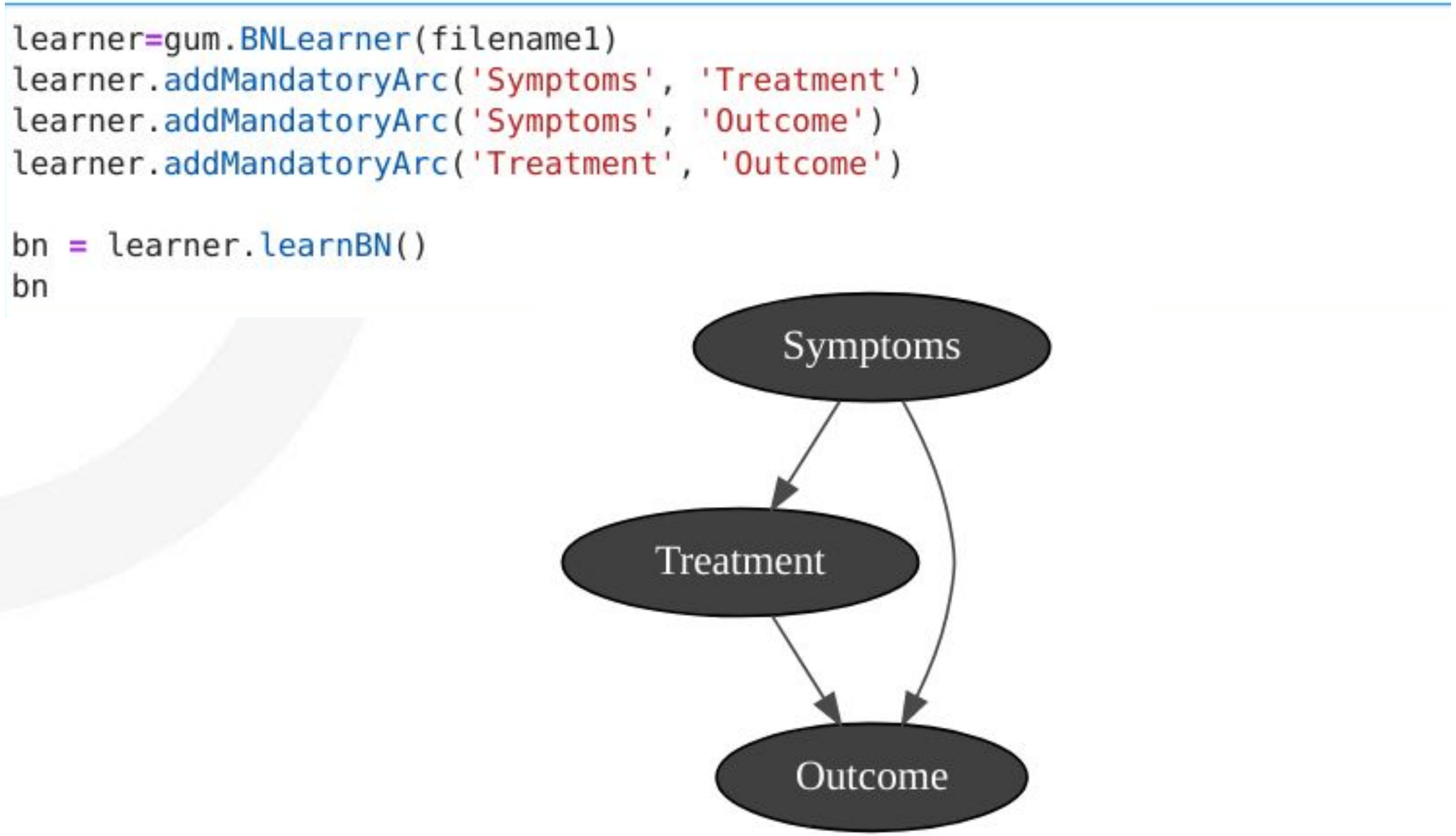
[2]:

	Treatment	Symptoms	Outcome
0	A	mild	alive
1	A	mild	alive
2	A	mild	alive
3	A	mild	alive
4	A	mild	alive
...	...	...	...
2045	A	mild	alive
2046	B	severe	alive
2047	A	mild	alive
2048	B	mild	alive
2049	A	mild	dead

2050 rows × 3 columns

Done with pyAgrum (2) : Learning

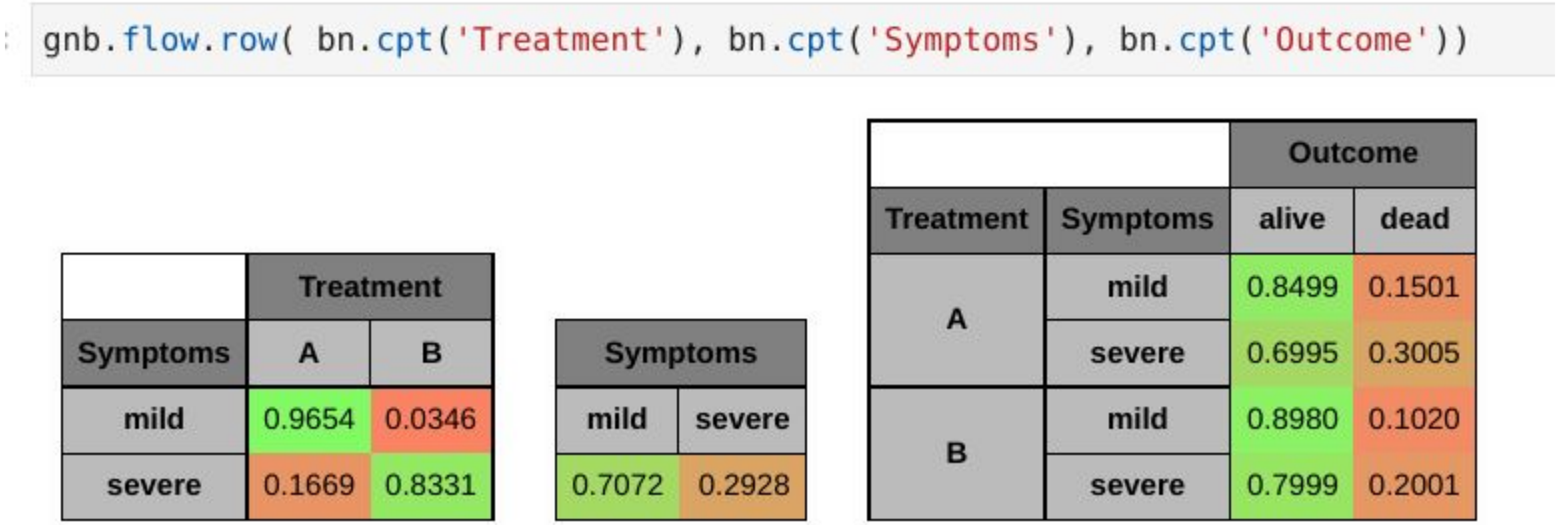
Learning model from data (using constraints)



```
learner=gum.BNLearner(filename1)
learner.addMandatoryArc('Symptoms', 'Treatment')
learner.addMandatoryArc('Symptoms', 'Outcome')
learner.addMandatoryArc('Treatment', 'Outcome')

bn = learner.learnBN()
bn
```

Done with pyAgrum (3): Inference

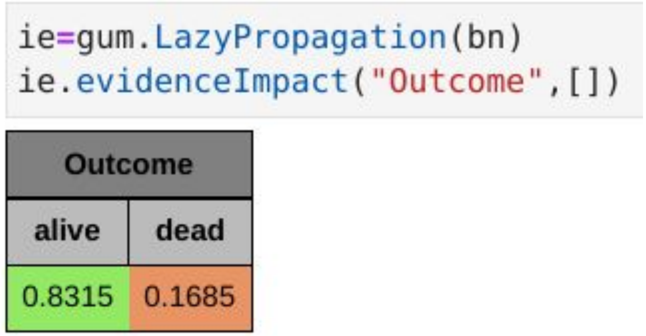


```
gnb.flow.row( bn.cpt('Treatment'), bn.cpt('Symptoms'), bn.cpt('Outcome'))
```

	Treatment	
Symptoms	A	B
mild	0.9654	0.0346
severe	0.1669	0.8331

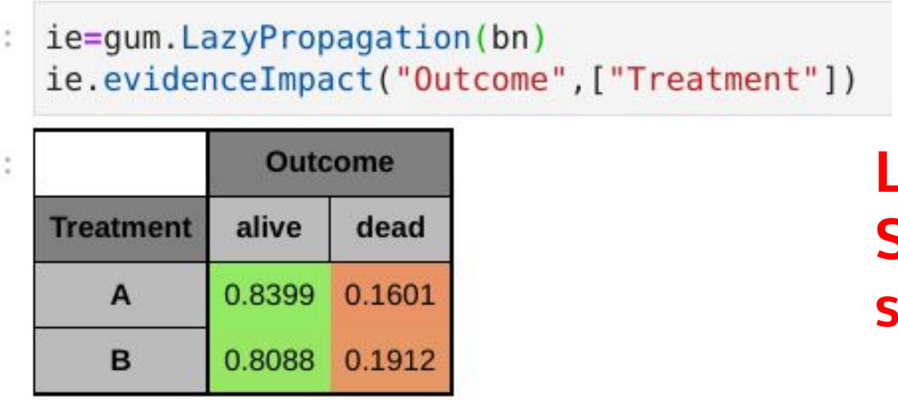
Symptoms	
mild	severe
0.7072	0.2928

		Outcome	
Treatment	Symptoms	alive	dead
A	mild	0.8499	0.1501
	severe	0.6995	0.3005
B	mild	0.8980	0.1020
	severe	0.7999	0.2001



```
ie=gum.LazyPropagation(bn)
ie.evidenceImpact("Outcome",[])
```

Outcome	
alive	dead
0.8315	0.1685



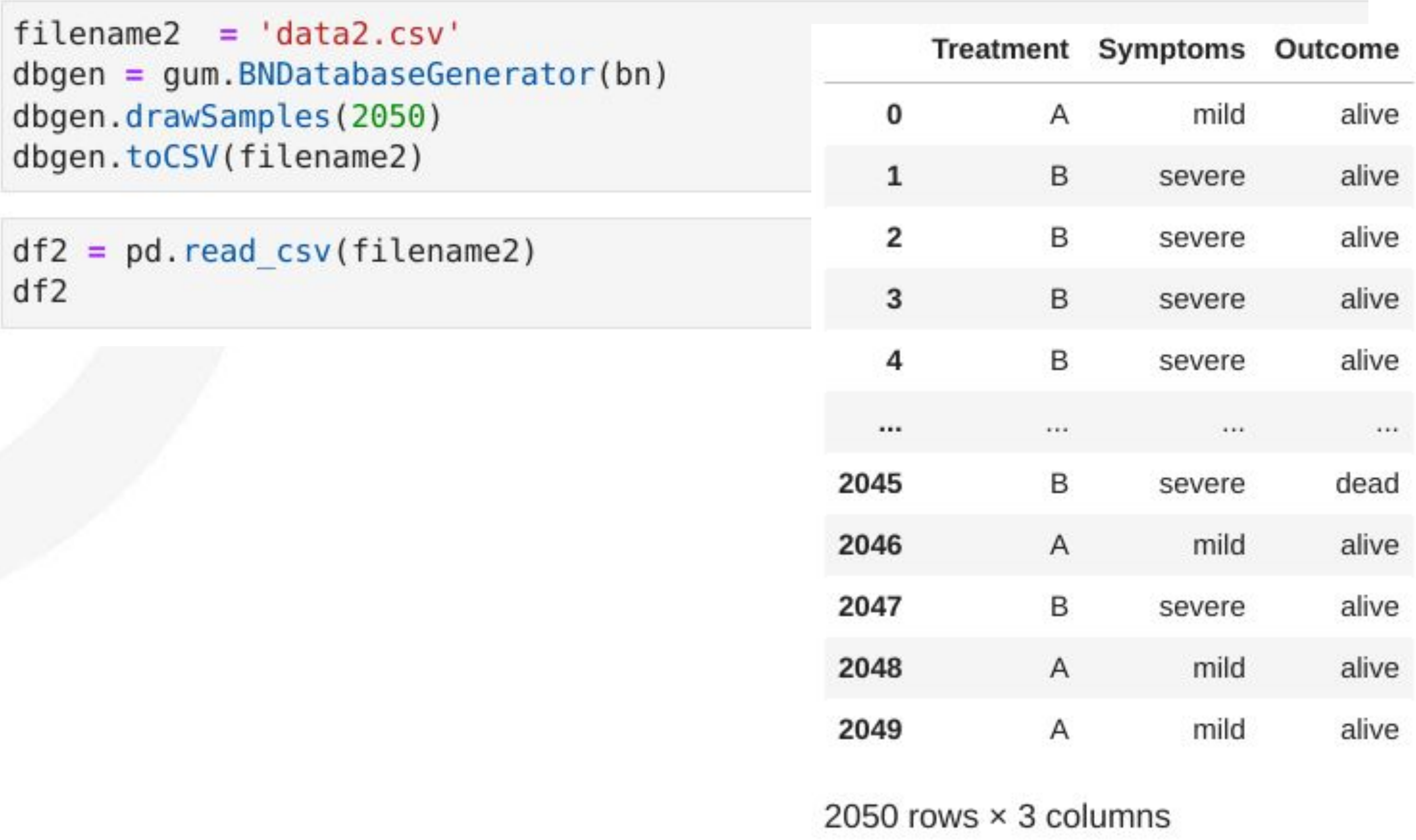
```
ie=gum.LazyPropagation(bn)
ie.evidenceImpact("Outcome",["Treatment"])
```

	Outcome	
Treatment	alive	dead
A	0.8399	0.1601
B	0.8088	0.1912

Look, that Simspon’s paradox still there !

Done with pyAgrum (4)

Generate (Draw) data from the model itself



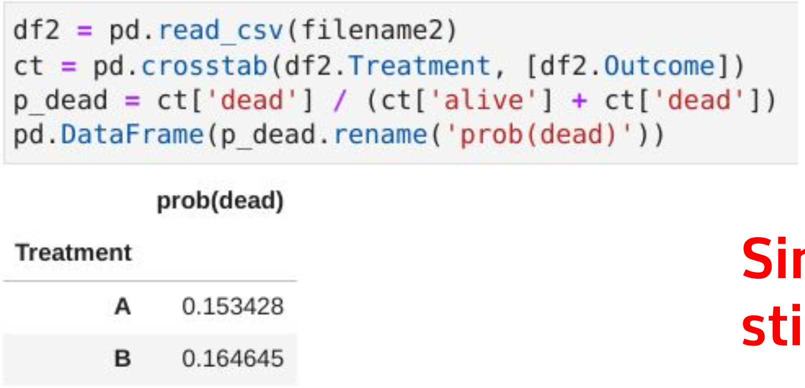
```
filename2  = 'data2.csv'
dbgen = gum.BNDatabaseGenerator(bn)
dbgen.drawSamples(2050)
dbgen.toCSV(filename2)
```

```
df2 = pd.read_csv(filename2)
df2
```

	Treatment	Symptoms	Outcome
0	A	mild	alive
1	B	severe	alive
2	B	severe	alive
3	B	severe	alive
4	B	severe	alive
...	...	...	...
2045	B	severe	dead
2046	A	mild	alive
2047	B	severe	alive
2048	A	mild	alive
2049	A	mild	alive

2050 rows × 3 columns

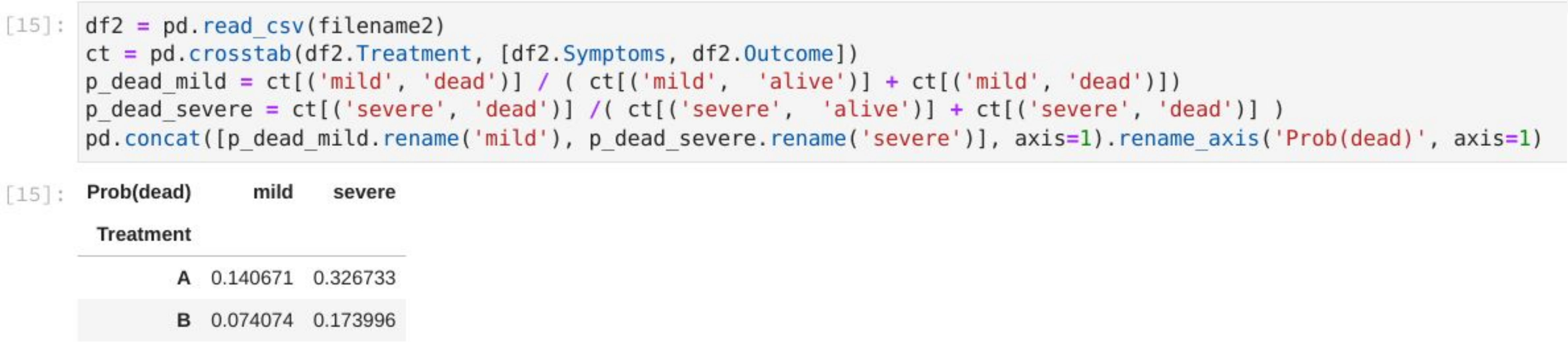
Done with pyAgrum (5)



```
df2 = pd.read_csv(filename2)
ct = pd.crosstab(df2.Treatment, [df2.Outcome])
p_dead = ct['dead'] / (ct['alive'] + ct['dead'])
pd.DataFrame(p_dead.rename('prob(dead)'))
```

	prob(dead)
Treatment	
A	0.153428
B	0.164645

Simspon’s paradox
still there !



```
[15]: df2 = pd.read_csv(filename2)
      ct = pd.crosstab(df2.Treatment, [df2.Symptoms, df2.Outcome])
      p_dead_mild = ct[('mild', 'dead')] / ( ct[('mild',  'alive')] + ct[('mild', 'dead')])
      p_dead_severe = ct[('severe', 'dead')] /( ct[('severe',  'alive')] + ct[('severe', 'dead')] )
      pd.concat([p_dead_mild.rename('mild'), p_dead_severe.rename('severe')], axis=1).rename_axis('Prob(dead)', axis=1)
```

[15]:

Prob(dead)	mild	severe
Treatment		
A	0.140671	0.326733
B	0.074074	0.173996

Done with pyAgrum (6)

Mild Symptoms

Severe Symptoms

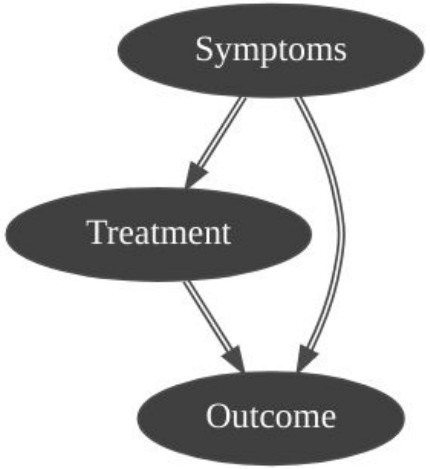
```
learner=gum.BNLearner(filename1)
learner.addMandatoryArc('Symptoms', 'Treatment')
learner.addMandatoryArc('Symptoms', 'Outcome')
learner.addMandatoryArc('Treatment', 'Outcome')

bn = learner.learnBN()
bn

d1 = csl.CausalModel(bn)
cslnb.showCausalImpact(d1, "Outcome", doing="Treatment",values={})
```

Treatment A

Treatment B



$$P(Outcome \mid \hookrightarrow Treatment) = \sum_{Symptoms} P(Outcome \mid Symptoms, Treatment) \cdot P(Symptoms)$$

	Outcome	
Treatment	**alive**	**dead**
A	0.8059	0.1941
B	0.8693	0.1307

Causal Model

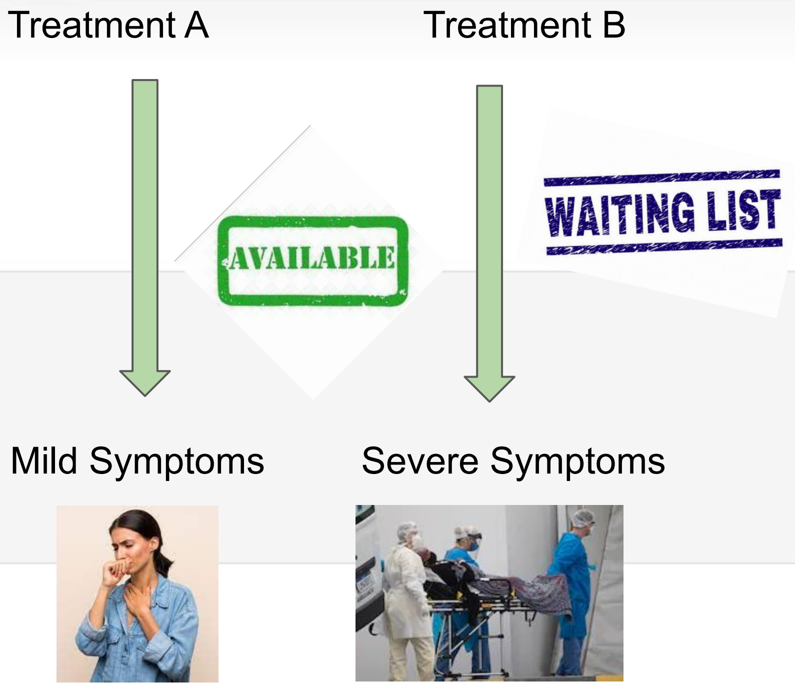
Explanation : backdoor ['Symptoms'] found.

Impact : $P(Outcome \mid \hookrightarrow Treatment)$

Done with pyAgrum (7)

```
learner=gum.BNLearner(filename1)
learner.addMandatoryArc('Treatment', 'Symptoms')
learner.addMandatoryArc('Symptoms', 'Outcome')
learner.addMandatoryArc('Treatment', 'Outcome')

bn2 = learner.learnBN()
d2 = csl.CausalModel(bn2)
cslnb.showCausalImpact(d2, "Outcome", doing="Treatment",values={})
```

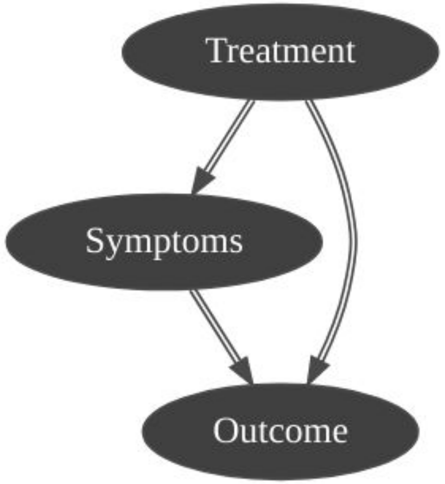


Treatment A

Treatment B

Mild Symptoms

Severe Symptoms



$$P(Outcome \mid \hookrightarrow Treatment) = \sum_{Symptoms} P\left(Outcome \mid Symptoms, Treatment\right) \cdot P\left(Symptoms \mid Treatment\right)$$

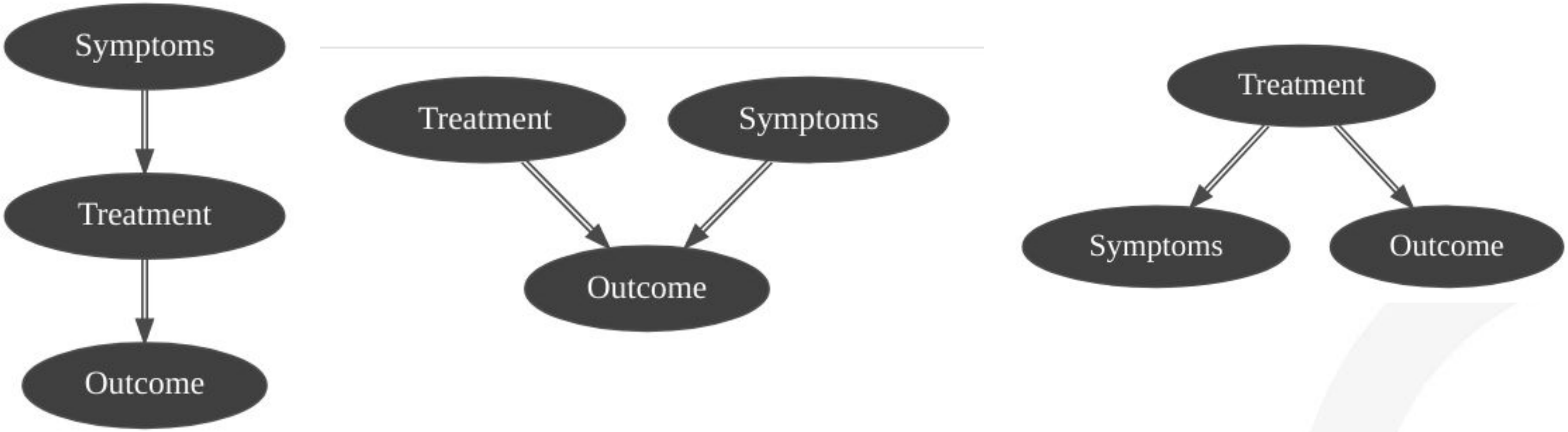
	Outcome	
Treatment	**alive**	**dead**
A	0.8399	0.1601
B	0.8088	0.1912

Causal Model

Explanation : Do-calculus computations

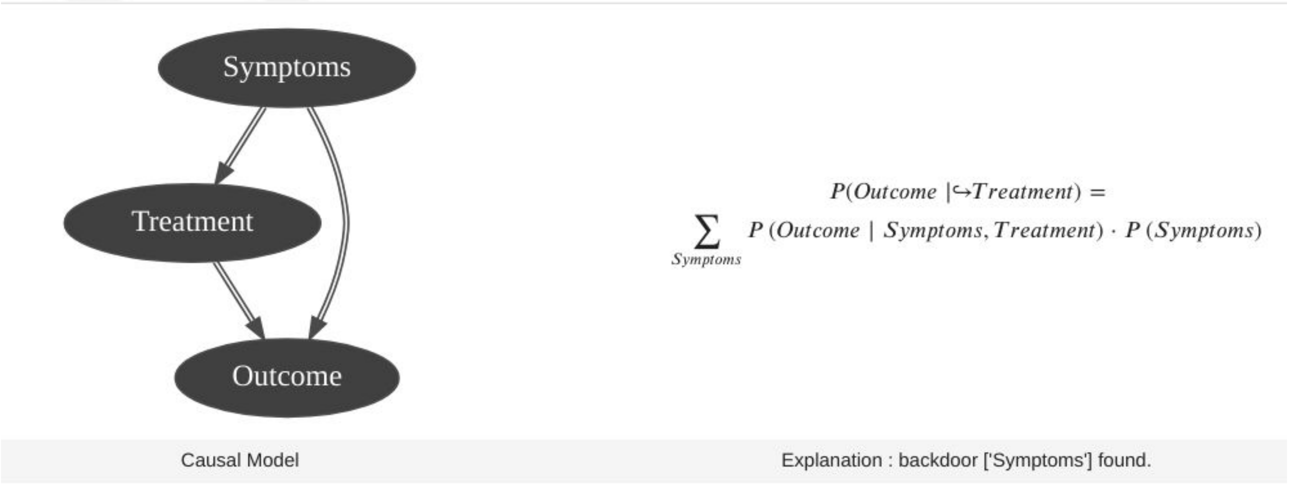
Impact : $P(Outcome \mid \hookrightarrow Treatment)$

Done with pyAgrum (8)



SOME CAUSAL DIAGRAMS (LIKE THESE) WON'T EVER GENERATE SIMPSON'S PARADOXES

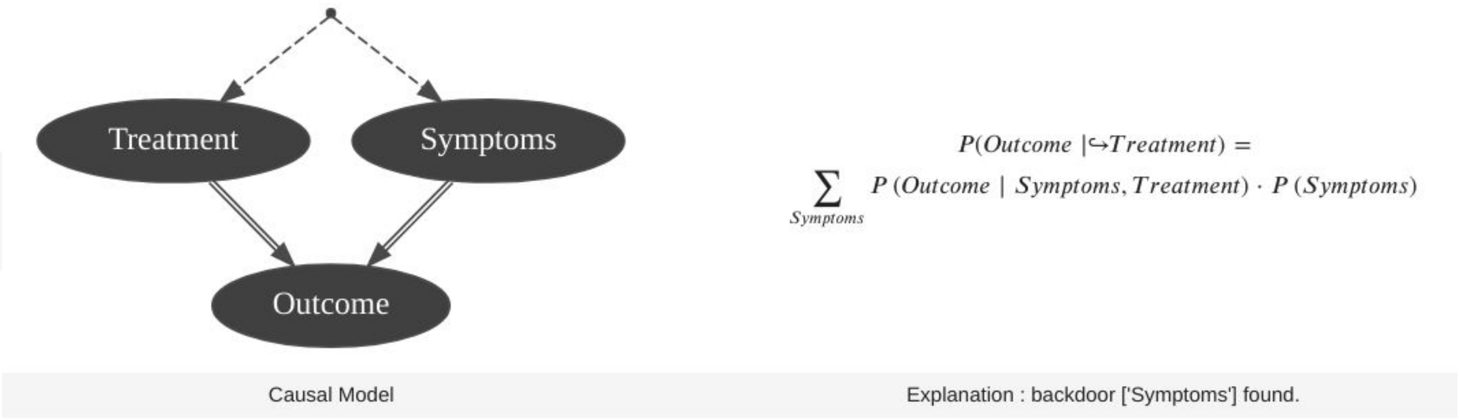
Done with pyAgrum (9) : Backdoors



$$P(Outcome \mid \hookrightarrow Treatment) = \sum_{Symptoms} P\left(Outcome \mid Symptoms, Treatment\right) \cdot P\left(Symptoms\right)$$

Causal Model

Explanation : backdoor ['Symptoms'] found.



$$P(Outcome \mid \hookrightarrow Treatment) = \sum_{Symptoms} P\left(Outcome \mid Symptoms, Treatment\right) \cdot P\left(Symptoms\right)$$

Causal Model

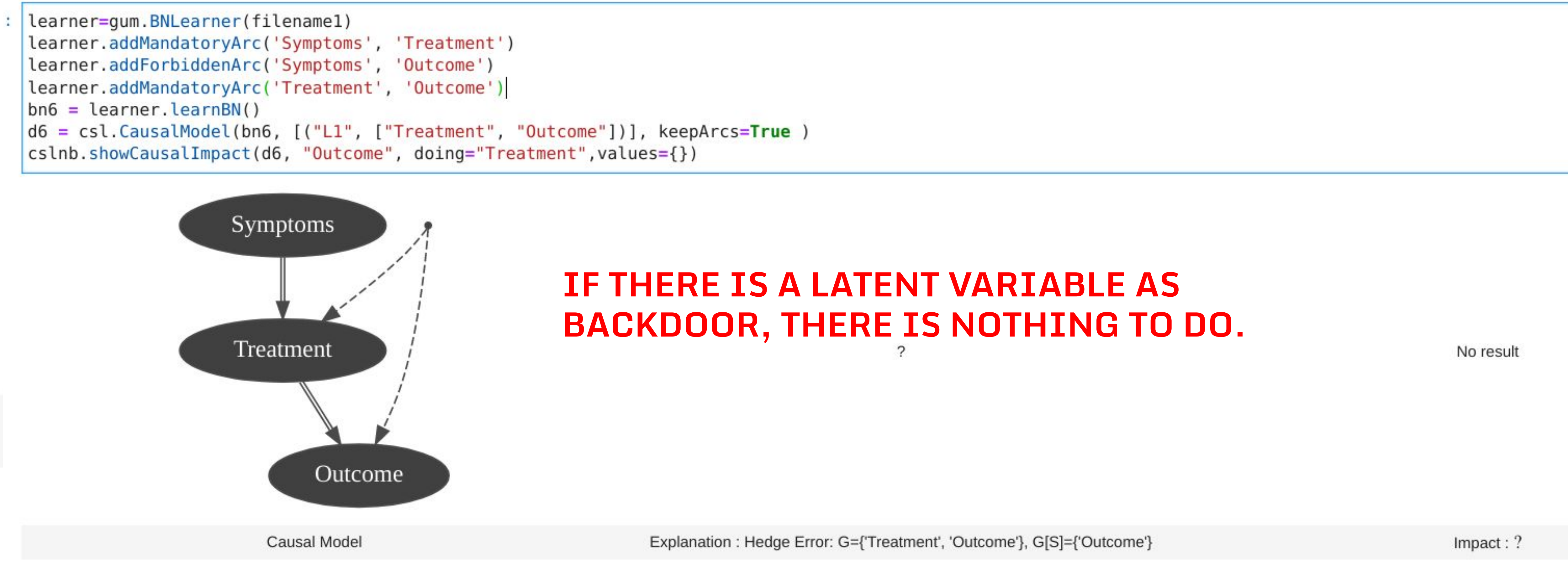
Explanation : backdoor ['Symptoms'] found.

IF THERE IS A BACKDOOR, YOU MUST CONDITION ON IT.

IF THERE IS A BACKDOOR CHOOSE B

	Outcome	
Treatment	alive	dead
A	0.8059	0.1941
B	0.8693	0.1307

Done with pyAgrum (10): Backdoors

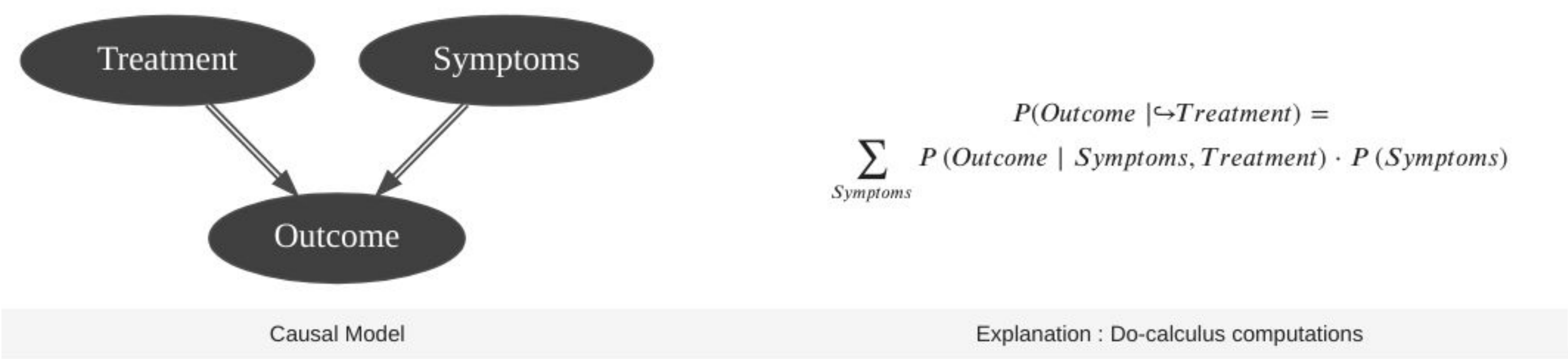


```
learner=gum.BNLearner(filename1)
learner.addMandatoryArc('Symptoms', 'Treatment')
learner.addForbiddenArc('Symptoms', 'Outcome')
learner.addMandatoryArc('Treatment', 'Outcome')
bn6 = learner.learnBN()
d6 = csl.CausalModel(bn6, [("L1", ["Treatment", "Outcome"])], keepArcs=True )
cslnb.showCausalImpact(d6, "Outcome", doing="Treatment",values={})
```

IF THERE IS A LATENT VARIABLE AS BACKDOOR, THERE IS NOTHING TO DO.

```
?                                                                 No result
Causal Model     Explanation : Hedge Error: G={'Treatment', 'Outcome'}, G[S]={'Outcome'}     Impact : ?
```

Done with pyAgrum (11): Backdoors



$$P(Outcome \mid\hookrightarrow Treatment) = \sum_{Symptoms} P\left(Outcome \mid Symptoms, Treatment\right) \cdot P\left(Symptoms\right)$$

Causal Model

Explanation : Do-calculus computations

$$P(Outcome \mid\hookrightarrow Treatment) = P\left(Outcome \mid Treatment\right)$$

Causal Model

Explanation : Do-calculus computations

IF THERE IS NO BACKDOOR, CHOOSE THE AGGREGATE.

Summary

When facing the Simpson's Paradox:

1. Draw a causal Diagram
2. If the is a backdoor
    a. If it is observed, the right decision it to use it as conditioning variable
    b. If it is unobserved, no conclusion can be made.
3. If there is no backdoor, the right decision is to use the aggregate result.

SCALNYX

BY DESIGN

Thank you.