COMP3224

Coursework I

Alberto Tamajo

Student ID: 30696844

February 2021
Electronics and Computer Science Department
University of Southampton

1 Exercise 1

1.1 Overview

Before delving into the details of this exercise and how its tasks have been solved, it is fundamental to lay out a theoretical framework to understand what colliders are and the conditional independence rule to which they are subject. Such a theoretical framework and conditional independence rule will be essential to explain the results of this exercise's tasks. Due to constraints on the number of pages for this report, the abovementioned theoretical framework and conditional independence rule have been included in the Appendices. It is highly recommended to read Appenidx A to appreciate the quality of this report and the rationale under which the tasks of this exercise have been carried out. Furthermore, Appendix A gives proof of deep understanding of the material covered in the lectures.

1.2 Exercise details and solutions

This exercise asks to implement a collider to demonstrate how selection bias can induce correlations between uncorrelated variables. A bunch of people are auditioning to be actors. The choice of cast C is the aggregation of scores A and B, where A and B are the actor's acting talent and the actor's perceived attractiveness, respectively. This audition process can be modelled through a Structural Causal Model $M = (U, V, F)$ where $U = \{U_A, U_B, U_C\}$, $V = \{A, B, C\}$ and $F = \{f_A = U_A, f_B = U_B, f_C = A + B + U_C\}$. The endogenous variable C is a collider in M because its function f_C is defined in terms of A and B. In other words, if we mapped M to its corresponding graph causal model G, then C would have two incoming edges. In this model, we do not try to explain the causes of the actor's acting talent and perceived attractiveness; this is the reason why the variables A and B only depend on the exogenous variables U_A, U_B. The variables U_A, U_B are assumed to be Standard Gaussian random variables, $U_A, U_B \sim \mathcal{N}(0, 1)$.

1.2.1 First task

The first task of this exercise requires us to draw scatter plots of (A, B), fit a regression line for A as a function of B and gain an understanding of what the slope of the regression line communicates about the association between A and B. By using the tools of statistics and the theoretical framework outlined in the Appendix A, it is possible to compute the slope of the regression line for A as a function of B and attain what it communicates about their association. Afterwards, we will see whether the slope of the regression line computed empirically matches the value of the slope calculated statistically.

Slope statistical analysis It turns out that the slope m of the regression line $A = mB + b + \epsilon$ is $m = \frac{Cov(B,A)}{Var(B)}$. Appendix B gives a detailed proof of the statistical derivation of the slope m. Thanks to the theoretical framework outlined in Appendix A, we know that we can map the Structural Causal Model M to a corresponding graph causal model G. If this is done, it is possible to notice that the only path between A and B includes the collider C. d-separation states that two variables are unconditionally independent if any path between them is blocked by a collider. Consequently, A and B are unconditionally independent variables. The covariance between two independent random variables is 0; this can be proved straightforwardly. To conclude, $m = \frac{Cov(B,A)}{Var(B)} = 0$. To answer the question posed by this exercise's task, the regression line between acting talent and attractiveness communicates that they are independent variables. At this point, it seems reasonable to spend a word of caution: if the slope of the best fitting line between two variables X and Y is 0, it does not imply that X and Y are independent. On the other hand, if X and Y are independent variables, the slope of the best fitting line necessarily needs to be 0. Given the fact that the variables A and B in this exercise are independent, then the slope of their best-fitting line needs to be 0.

Slope empirical analysis Figure 1 in Appendix C shows several scatter plots of $(A, B) = (a_n, b_n), n = 1, ..., N$ for different values of N. As it is possible to notice, the slope of the best-fitting line for A as a function of B tends towards 0 as the value of N increases. The cause behind this phenomenon can be explained through the Law of Large Numbers(LLN): let $X_1, X_2, ..., X_N$ be i.i.d random variables sampled from a given distribution X. It follows that $\lim_{N\to\infty} \sum_{i=1}^{N} \frac{X_i}{N} - \mathbb{E}(X) = 0$. In other words, as the sample size increases, the sample mean tends towards the real expected value. In the context of this exercise, the Law of Large Numbers tells us that given $Z = (A - \mathbb{E}(A))(B - \mathbb{E}(B)))$ and $Z_1, Z_2, ..., Z_N$ be i.i.d random variables sampled from Z, then $\lim_{N\to\infty} \sum_{i=1}^{N} \frac{Z_i}{N} - \mathbb{E}(Z) = 0 \to \lim_{N\to\infty} \sum_{i=1}^{N} \frac{Z_i}{N} = 0$ because in the previous section we have proved that $\mathbb{E}(Z) = 0$. Therefore, the empirical results seem to match the statistical solution given in the paragraph above as the larger the number of samples, the more the slope of the best-fitting line for A as a function of B tends to 0.

1.3 Second task

The second task of this exercise consists of setting a positive threshold θ for C, selecting all the points $D = (a_n, b_n)$ that lie above such threshold, plotting them on a scatterplot and doing a linear regression for B against A for the selected data D. In the previous task, we have discovered that A and B are unconditionally independent, $P(A, B) = P(A)P(B)$. The conditional independece rule in colliders states that given two causes X and Y, which are unconditionally independent, they become dependent when conditioned on the common effect Z. Therefore, even though A and B are unconditionally independent, they become dependent when conditioned on C, $P(A, B|C) \neq P(A|C)P(B|C)$. Therefore, given that $D = (a_n, b_n)$ contains all points that lie above the threshold θ, the regression line for B against A will show that B and A are correlated. Since, $C = B + A$, it is possible to state right away that B and A are negatively correlated because if one of the variables increases in value, the other variable needs to balance that effect decresing its value. Given that B and A are negatively correlated, it follows that the slope m of the best-fitting line for B against A needs to be negative as $m = \frac{Cov(A,B)}{Var(A)}$. Figure 2 in Appendix D illustrates several scatter plots for different Ds and the corresponding regression lines for B against A. In conclusion, given that we condition on the choice of cast, the actor's acting talent and perceived attractiveness become dependent.

2 Exercise 2

Exercise 2 involves generating samples from a conditional probability distribution for binary variables A and B. The aim of this exercise is to show empirically that it is possible to draw samples from a joint probability distribution using conditional probabilities. The mathematical validity of the previous statement is proved in the Appendix E. Once again, it is highly recommended to read the Appendices to appreciate the quality of this report which gives proof of deep understanding of the material covered in the lectures.

In order to show empirically that drawing samples from a joint probability distribution can be performed using conditional probabilities, the following probabilities are going to be used:

$$P(A = 0) = 0.6 \qquad P(B = 0|A = 0) = 0.45 \qquad P(B = 0|A = 1) = 0.75$$

By using the second Kolmogorov axiom, the remaining probabilities are:

$$P(A = 1) = 0.4 \qquad P(B = 1|A = 0) = 0.55 \qquad P(B = 1|A = 1) = 0.25$$

Using the equation $P(A, B) = P(B|A)P(A)$, the joint distribution of A and B is characterised by the following values:

$$P(A = 0, B = 0) = 0.27 \qquad P(A = 0, B = 1) = 0.33 \qquad P(A = 1, B = 0) = 0.3 \qquad P(A = 1, B = 1) = 0.1$$

Figure 3 in Appendix E illustrates the Python code used to generate pairs (a, b) from the above joint distribution using conditional probabilities. Firstly, a is sampled from $P(A)$, subsequently b is sampled from the conditional distribution $P(B|A = a)$. The function "infer_joint_distribution" computes the proportion of times each pair (a, b) occurs given N samples. Thanks to the Law of Large Numbers, we expect the frequency of each pair (a, b) to tend towards $P(A = a, B = b)$ as N gets larger and larger. This is exactly what happens; therefore, the empirical results also demonstrate that drawing samples from a joint distribution can be performed using conditional probabilities.

3 Exercise 3

Study question 1.5.2 of "Causal Inference in Statistics: A Primer" proposes a graph causal model and this exercise asks the following question, given the following probabilites:

$$P(Z = z_1) = r \quad P(X = x_1|Z = z_0) = q_1 \quad P(X = x_1|Z = z_1) = q_2 \quad P(Y = y_1|Z = z_0, X = x_0) = p_1$$

$$P(Y = y_1|Z = z_0, X = x_1) = p_2 \quad P(Y = y_1|Z = z_1, X = x_0) = p_3 \quad P(Y = y_1|Z = z_1, X = x_0) = p_4$$

is it possible to generate samples drawn from the joint distribution $P(X, Y, Z)$? The answer is positive and a small proof is provided: given that $P(A, B) = P(B|A)P(A)$, it follows that $P(X, Y, Z) = P(Y|Z, X)P(Z, X) = P(Y|Z, X)P(X|Z)P(Z)$. Therefore, this exercise, like the previous one, tells us that we can generate samples from a joint distribution using conditional distributions. In the specific case of this exercise, we can draw samples from $P(X, Y, Z)$ using the following two mechanisms:

- Draw (x, y, z) tuples directly from $P(X, Y, Z)$
- Draw z from $P(Z)$, subsequently draw x from $P(X|Z = z)$ and lastly draw y from $P(Y|Z = z, X = x)$

The two abovementioned mechanisms to generate samples from a joint distribution are perfectly equivalent, and the latter is implemented in Python, as illustrated in Figure 4 in Appendix F. By instantiating the values $r, q_1, q_2, p_1, p_2, p_3, p_4$ and using the "draw_from_joint" function in Figure, it is possible to notice that the proportion of times each tuple (x, y, z) occurs given N samples, approaches $P(X = x, Y = y, Z = z)$ as N gets larger and larger. Thus, this empirical results confirm that it is possible to draw samples from a joint distribution using conditional distributions. Appendix G introduces the rule of product decomposition by showing that the joint distribution $P(X, Y, Z) = P(Y|Z, X)P(Z, X) = P(Y|Z, X)P(X|Z)P(Z)$ can be rewritten differently if the graph causal model in study question 1.5.2 is taken into consideration.

Appendices

A Colliders theoretical framework and conditional independence rule

In order to understand the nature of colliders, firstly, it is essential to introduce the concept of Structural Causal Models(SCMs).

A.1 Structural Causal Models

A Structural Causal Model(SCM) is a model that describes the causal mechanism through which a certain data set has been generated. In other words, a SCM tries to model all the relevant variables that participate in a given phenomenon and how they interact with each other. Given that the causal mechanism behind a data set is known, it is possible to generate new data which exhibits the same patterns. Thus, SCM may be used to simulate new data as well. Mathematically, a SCM is a three-tuple (U, V, F) where U is the set of exogenous variables, V is the set of endogenous variables, and F is the set of functions that assign each variable in V a value based on the values of the other variables in the model. The set of exogenous variables U contains all those variables whose causes are not explained by the model. Such causes may be irrelevant to the problem or even unknown. Exogenous variables are crucial in a SCM as they stand in for any unknown or random effects that may alter the relationship between the endogenous variables. This holds because, in most natural phenomena, we only know a subset of causes for a given variable(the endogenous variables). Thus, to model the remaining subset of unknown causes, we need to fall back to the exogenous variables, which are commonly treated as random variables with an underlying distribution. On the other hand, the endogenous variables' causes are delineated by the model. These causes may either be exogenous variables, endogeneous variables or a combination of both. If the value of every exogenous variable in a SCM is known, the functions in F enable the determination of each endogenous variable. By means of Structural Causal Models, it is possible to define causation formally: **a variable X is a direct cause of a variable Y if Y's function is directly defined in terms of X. X is a cause of Y if it is a direct cause of Y, or of any cause of Y.**
It turns out that every SCM has an associated graphical causal model; the next section will give more details about such models.

A.2 Graph causal models

A graph causal model is a directed graph (L, E) that graphically models the relevant variables that participate in a given phenomenon and how they interact with each other. Every SCM can be mapped to a graph causal model through a function $T : (U, V, F) \to (L, E)$, where (L, E) denotes a graph with vertices L and edges E. The function $T(\cdot)$ is defined as follows:

$$T(U, V, F) = (U \cup V, \{(x, y) | x \text{ is an argument of the function} f_y \in F)\}$$

Substantially, given as input a SCM M, the function $T(\cdot)$ maps M to a graph where the nodes represent the exogenous and endogenous variables and the directed edges between the nodes represent the functions in F. It is important to notice that $T(\cdot)$ is not injective; thus, multiple SCMs have the same underlying graph causal model. Because of the relationship between SCMs and graph models, we can give a graph definition of causation: **if in a graph model, the variable X is the parent of a variable Y, X is a direct cause of Y. Similarly, if the variable X is an ancestor of the variable Y, X is a likely cause of Y.**
If X is an ancestor of a variable Y, it does not necessarily imply that X is a cause of Y because the causation relation is not transitive. Therefore, X is said to be a likely cause of Y. Given the above definition of graph causation, it follows that the exogenous variables are root nodes in a graph causal model. Although graph causal models are less informative than SCMs as they lack the quantitative aspect of causal relationships, they provide an intuitive understanding of the patterns of dependencies among the variables. Therefore, we can learn which variables in the data set are independent of each other and which are independent of each other conditional on other variables by merely looking at their graph causal model. The tool that allows understanding the dependency relations in a graph causal model of any complexity is called **d-separation**. d-separation builds upon a set of three rules that are derived from a set of three special graph causal models: chains, forks and colliders.
In this Appendix, chains and forks will not be described as they are not essential to understand Exercise 1. In contrast, the next section will outline the definition of colliders and their conditional independence rule.

A.3 Colliders

Let $G = (\{U_x, U_y, U_z, X, Y, Z\}, \{(U_x, X), (U_y, Y), (U_z, Z), (X, Z), (Y, Z)\})$, then G is the simplest graph causal model containing a collider. Essentially, a collider occurs when one node has incoming edges from other two nodes which are not linked by an edge. By taking the abovementioned graph G as a reference, the dependency relations in a collider are as follows:

- X and Z are dependent
- Y and Z are dependent
- X and Y are independent
- X and Y are dependent conditioned on Z

The first two statements are easy to prove; both X and Y are parents of the node Z, so they are direct causes of Z. Consequently, the dependency relation needs to hold necessarily. Proving the third statement is also straightforward as neither X nor Y is a descendant or an ancestor of the other. The fourth statement states that even though X and Y are independent, they become dependent if conditioned on their common effect Z. At first, this statement may look surprising, and the reason behind this lies in the fact that we tend to associate dependence with causation. However, colliders prove that dependence may also arise when two variables are conditioned on the same effect, violating the assumption of **"no correlation without causation"**. The reason why the fourth statement is true relies on the fact that that Z depends, for its value, on X and Y. Therefore, if we keep the value of Z fixed, any change in the value of Y needs to be compensated by a corresponding change in the value of X and vice versa. Statement 4 can be summarised more formally through the Conditional Independence rule in Colliders: **If there is only one path between X and Y, and Z is the collision node between X and Y, then X and Y are independent but become dependent if conditioned on and any descendants of Z.**

B Slope statistical analysis

The following equations will help deriving the slope of the best fitting line $A = mB + b + \epsilon$:

$$
\begin{aligned}
&\mathbb{E}(A) = 0 && (1)\\
&\mathbb{E}(B) = 0 && (2)\\
&\mathbb{E}(\epsilon) = 0 && (3)\\
&Cov(A, B) = \mathbb{E}[(A - \mathbb{E}(A))(B - \mathbb{E}(B))] = \mathbb{E}(AB) && (4)\\
&Var(B) = \mathbb{E}[(B - \mathbb{E}(B))^2] = \mathbb{E}(B^2) && (5)\\
&Cov(B, \epsilon) = \mathbb{E}(B\epsilon) = 0 && (6)
\end{aligned}
$$

The first two equations come from the fact that both A and B are mean centred random variables. ϵ is the noise term and is assumed to be mean centred without loss of generality. The fourth and fifth equations follow directly from the first two equations. The last equation comes from the orthogonality principle, which states that the best least-square fitting line is obtained when the noise term ϵ is uncorrelated with each of the regressors. The slope and intercept of the best fitting line $A = mB + b + \epsilon$ are the following:

$$
\begin{aligned}
A &= mB + b + \epsilon && (7)\\
\Rightarrow \mathbb{E}(A) &= \mathbb{E}(mB + b + \epsilon) && (8)\\
&= m\,\mathbb{E}(B) + \mathbb{E}(b) + \mathbb{E}(\epsilon) && (9)\\
\Rightarrow b &= 0 && (10)\\
\Rightarrow A &= mB + \epsilon && (11)\\
\Rightarrow BA &= B(mB + \epsilon) && (12)\\
&= mB^2 + B\epsilon && (13)\\
\Rightarrow \mathbb{E}(BA) &= \mathbb{E}(mB^2 + B\epsilon) && (14)\\
&= m\,\mathbb{E}(B^2) + \mathbb{E}(B\epsilon) && (15)\\
\Rightarrow Cov(B, A) &= mVar(B) + Cov(B, \epsilon) && (16)\\
&= mVar(B) && (17)\\
\Rightarrow m &= \frac{Cov(B, A)}{Var(B)} && (18)\\
& && (19)
\end{aligned}
$$

C Slope empirical analysis

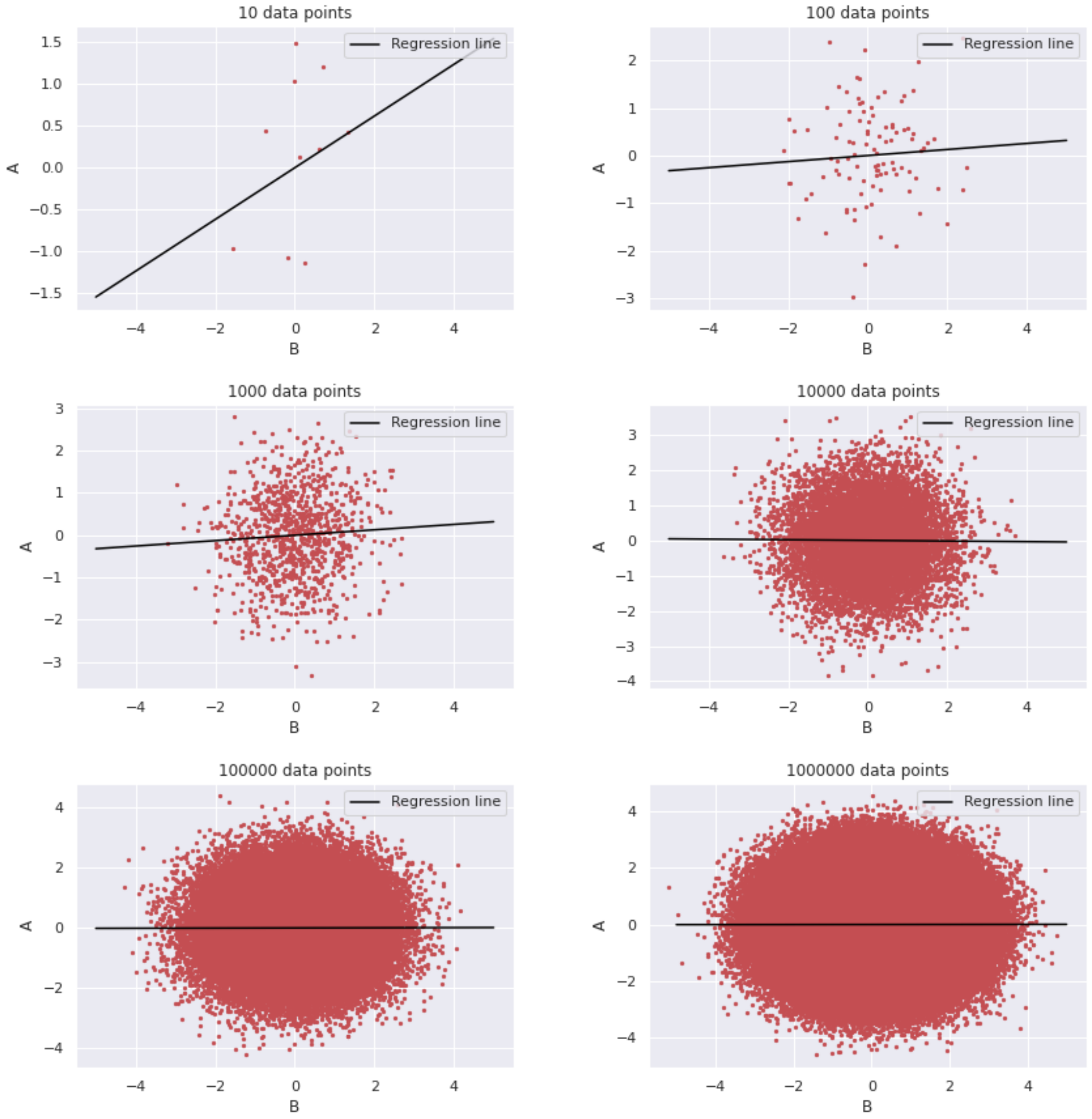


Figure 1: Scatter plots of $(A, B) = (a_n, b_n), n = 1, ..., N$ for different values of N and their corresponding regression lines for A as a function of B.

D Slope with threshold empirical analysis

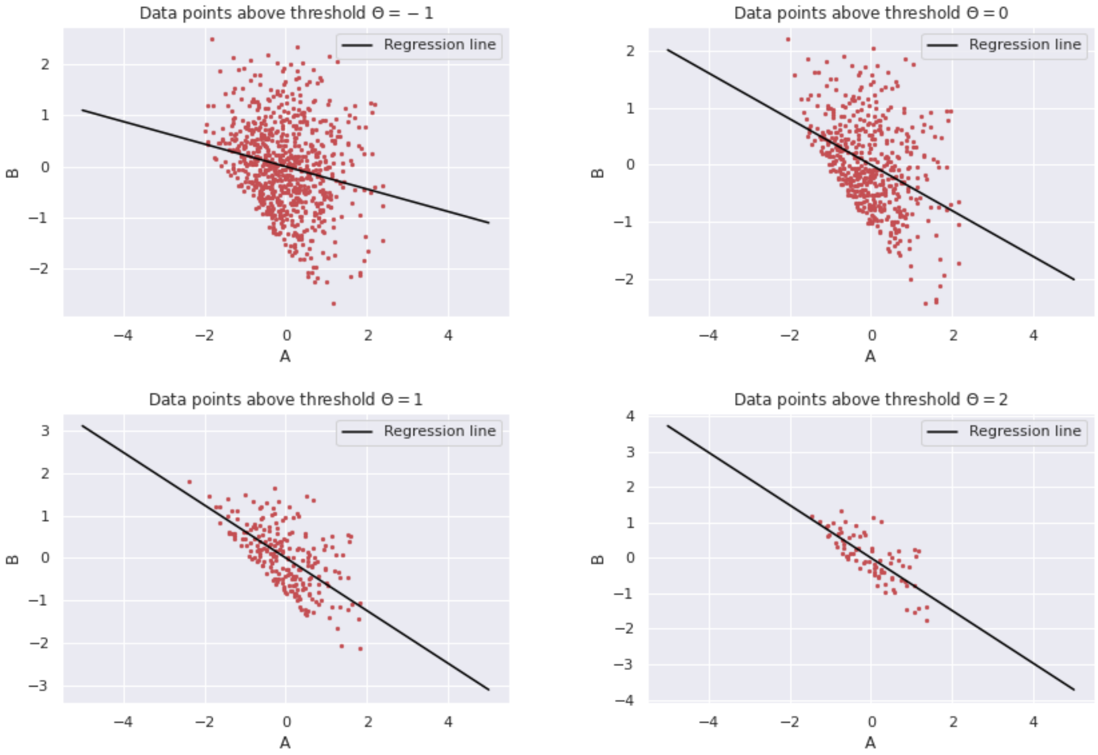


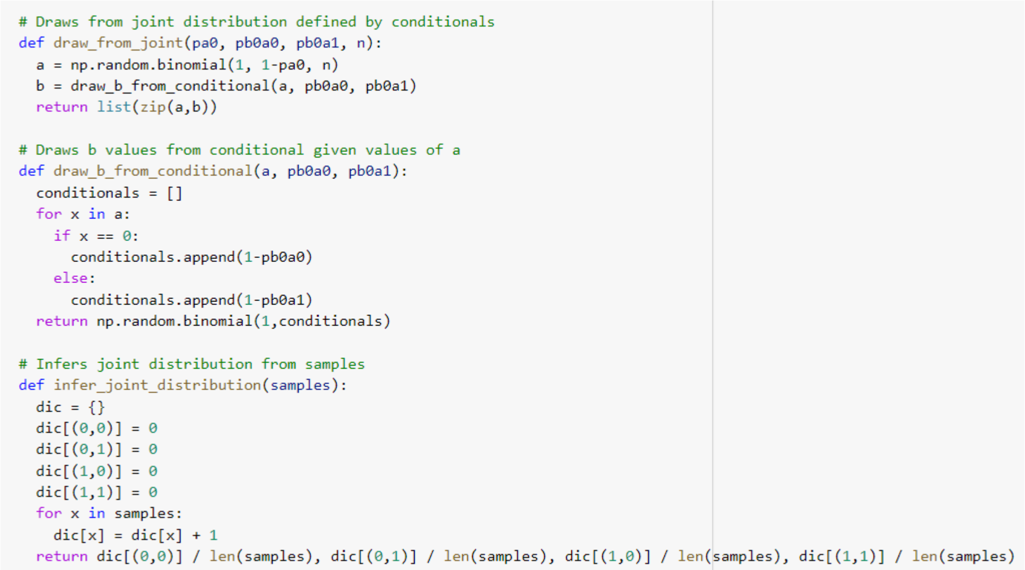
Figure 2: Scatter plots for different $D = \{(a_n, b_n)|a_n + b_n > \theta\}$ and the corresponding regression lines for B against A.

E Drawing samples from a joint probability distribution using conditional probabilities

Without a loss of generality, let A, B be two random variables and S_A, S_B be their sample space, respectively. The joint probability distribution of A and B, denoted as $P(A, B)$, describes the probability distribution over all possible pairs (a, b) where $a \in S_A$ and $b \in S_B$. The joint distribution $P(A, B)$ encodes every information about the random variables A and B. Indeed, the marginal distributions for each variable and the conditional probability distributions can be derived from the joint distribution. The conditional probability of B given A, denoted as $P(B|A)$, measures the probability of an event $b \in S_B$ occurring given that another event $a \in S_A$ has already occurred. Simply put, the conditional probability of B given A measures the fraction of probability A that intersects with B. Consequently, $P(B|A) = \frac{P(A,B)}{P(A)}$, which can be rearranged as follows: $P(A, B) = P(B|A)P(A)$. The latter equation tells us that we can generate samples from a joint distribution in two ways:

- Draw (a, b) pairs directly from $P(A, B)$
- Draw a from $P(A)$ and subsequently draw b from $P(B|A = a)$

The abovementioned mechanisms to generate samples from a joint distribution are perfectly equivalent. The second one of them is implemented in Figure 3



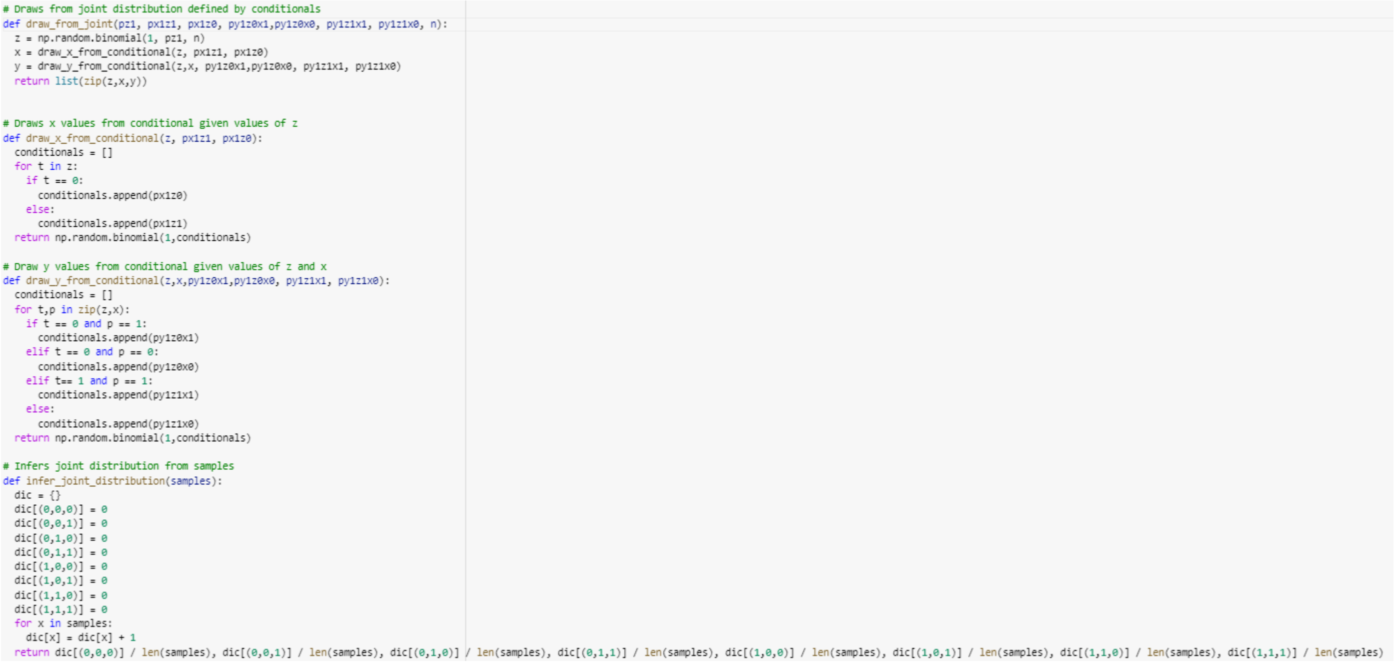
```
# Draws from joint distribution defined by conditionals
def draw_from_joint(pa0, pb0a0, pb0a1, n):
  a = np.random.binomial(1, 1-pa0, n)
  b = draw_b_from_conditional(a, pb0a0, pb0a1)
  return list(zip(a,b))

# Draws b values from conditional given values of a
def draw_b_from_conditional(a, pb0a0, pb0a1):
  conditionals = []
  for x in a:
    if x == 0:
      conditionals.append(1-pb0a0)
    else:
      conditionals.append(1-pb0a1)
  return np.random.binomial(1,conditionals)

# Infers joint distribution from samples
def infer_joint_distribution(samples):
  dic = {}
  dic[(0,0)] = 0
  dic[(0,1)] = 0
  dic[(1,0)] = 0
  dic[(1,1)] = 0
  for x in samples:
    dic[x] = dic[x] + 1
  return dic[(0,0)] / len(samples), dic[(0,1)] / len(samples), dic[(1,0)] / len(samples), dic[(1,1)] / len(samples)
```

Figure 3: Generative Python code for Exercise 2

F Study question 1.5.2 generative code



```
# Draws from joint distribution defined by conditionals
def draw_from_joint(pz1, px1z1, px1z0, py1z0x1,py1z0x0, py1z1x1, py1z1x0, n):
  z = np.random.binomial(1, pz1, n)
  x = draw_x_from_conditional(z, px1z1, px1z0)
  y = draw_y_from_conditional(z,x, py1z0x1,py1z0x0, py1z1x1, py1z1x0)
  return list(zip(z,x,y))


# Draws x values from conditional given values of z
def draw_x_from_conditional(z, px1z1, px1z0):
  conditionals = []
  for t in z:
    if t == 0:
      conditionals.append(px1z0)
    else:
      conditionals.append(px1z1)
  return np.random.binomial(1,conditionals)

# Draw y values from conditional given values of z and x
def draw_y_from_conditional(z,x,py1z0x1,py1z0x0, py1z1x1, py1z1x0):
  conditionals = []
  for t,p in zip(z,x):
    if t == 0 and p == 1:
      conditionals.append(py1z0x1)
    elif t == 0 and p == 0:
      conditionals.append(py1z0x0)
    elif t== 1 and p == 1:
      conditionals.append(py1z1x1)
    else:
      conditionals.append(py1z1x0)
  return np.random.binomial(1,conditionals)

# Infers joint distribution from samples
def infer_joint_distribution(samples):
  dic = {}
  dic[(0,0,0)] = 0
  dic[(0,0,1)] = 0
  dic[(0,1,0)] = 0
  dic[(0,1,1)] = 0
  dic[(1,0,0)] = 0
  dic[(1,0,1)] = 0
  dic[(1,1,0)] = 0
  dic[(1,1,1)] = 0
  for x in samples:
    dic[x] = dic[x] + 1
  return dic[(0,0,0)] / len(samples), dic[(0,0,1)] / len(samples), dic[(0,1,0)] / len(samples), dic[(0,1,1)] / len(samples), dic[(1,0,0)] / len(samples), dic[(1,0,1)] / len(samples), dic[(1,1,0)] / len(samples), dic[(1,1,1)] / len(samples)
```

Figure 4: Generative Python code for study question 1.5.2

G Product decomposition

In Exercise 3, it has been shown that $P(X, Y, Z) = P(Y|Z, X)P(X|Z)P(Z)$. By looking at the graph causal model in study question 1.5.2, the latter equation can be rephrased as $P(X, Y, Z) = P(Y|pa_Y)P(X|pa_X)P(Z|pa_Z)$, where pa_N denotes the set of all parents of a given node N. It would be nice if the above property were to hold for any given graph causal model. In other words, given any graph causal model $G(\{X_1, X_2, ..., X_N\}, E)$, $P(X_1, X_2, ..., X_N) = \prod_{i=1} P(X_i|pa_i)$. It turns out that the above property, called Rule of Product Decomposition, actually holds for any given graph causal model given that it is a Directed Acyclic Graph(DAG). The proof behind the Rule of Product Decomposition relies on the Conditional Independence Rule in Chains, Conditional Independence Rule in Forks, and Conditional Independence Rule in Colliders. Therefore, a further advantage of graph causal models, which was not mentioned in Appendix A, is that they enable us to express joint distributions very efficiently. Indeed, assuming that a graph causal model consists of N binary random variables and each variable is influenced by at most k other variables, the joint probability distribution can be specified by $N2^k$ numbers; this is quite an improvement given that, without a graph causal model, 2^N numbers are needed to specify the joint distribution. This is a general property of locally structured systems where each subcomponent interacts directly with only a bounded number of other components, regardless of the total number of components. Looking back at the Intelligent Systems module, it is evident that graph causal models are Bayesian Networks. However, it is of paramount importance to state that not all Bayesian Networks are graph causal models. The previous statement holds because Bayesian Networks do not constraint the construction of the DAG to be under causal knowledge. As an example, a Bayesian Network may be built using diagnostic knowledge rather than relying on causal knowledge. This module teaches us that while causal knowledge is robust, diagnostic knowledge is not. In conclusion, graph causal models are a subset of Bayesian Networks.