CORRELATION, MUTUAL INFORMATION AND NEURAL NETWORKS

Oona Rainio

MSc Thesis
September 2021

DEPARTMENT OF MATHEMATICS AND STATISTICS

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

RAINIO, OONA: Correlation, mutual information and neural networks
MSc Thesis, 67 pages, 11 appendix pages, 18 figures
Statistics
September 2021

This Master's thesis focuses on different measures of dependence. To study correlation, we introduce not only Pearson's, Spearman's and Kendall's correlation coefficients but also the maximal correlation coefficient. We consider the concept of mutual information derived from Shannon's information theory and the related information coefficient of correlation. Furthermore, we research a newer non-parametric quantity, the maximal information coefficient, about whose usability there have been conflicting views.

We first introduce the known properties of these measures from the literature and then check how well they work. For instance, we study how the exact type of dependence, the amount of statistical noise and the number of observations affect the performance of these coefficients. We are interested in finding such a quantity that effectively recognizes the dependence between two variables, regardless of if this relationship is linear, non-linear but monotonic, non-monotonic but functional, or non-functional.

To compute the values of these measures of dependence, we mostly use the programming language R and its newly developed packages with functions designed for this exact purpose. We also introduce a recent neural estimation algorithm MINE implemented within the PyTorch library of Python. We consider here both simulated data with several distinct types of dependence and real data from a few specific topics, such as the weather, youth behavior and air pollution.

Keywords: Correlation, maximal correlation, maximal information coefficient, measures of dependence, mutual information, neural networks.

Contents

1 Introduction

In the reality surrounding us, there are countless different variables somehow connected to each other. We study their relationships to understand the world better, find new information and predict the future. Throughout the history of statistics, several different tools have been developed to describe distinct types of dependence in a way that would fit the requirements of the time.

In the 19th century, Pearson's correlation coefficient was introduced as a simple indicator of linear dependence between two variables. During the 1900s, this concept was extended for measuring non-linear but monotonic dependence by Spearman and Kendall, and later the maximal correlation was created for recognizing non-monotonic relationships. In the 1940s, the birth of information theory brought forth the notion of entropy for expressing how much information one variable gives about itself, which also led the formulation of mutual information in order to study the information conveyed between the variables. In 2011, a yet another new quantity, the maximal information coefficient, was proposed for finding interrelated variables in the constant stream of digitized data.

This raises the question how one knows which of these concepts suggested during the last 150 years should be used in a certain situation. Since each measure of dependence was created to serve in the best possible way within the conditions of that time, these quantities have different properties. The newest of them is not necessarily the best choice for studying relationships for otherwise the simpler correlation coefficients would not still be so commonly used in the research of today. Furthermore, because of the incredible development of information technology since the introduction of the first correlation coefficient, the profound change in the computational methods also needs to be taken into account when comparing these measures of dependence.

However, while there is a lot of research about each correlation coefficient and the other quantities mentioned, there is not so much direct comparison between these measures of dependence or their computation. Namely, most of the scientific articles seem to focus on only one coefficient and its properties without considering such circumstances in which some other option would work better. The research of these coefficients also relies very much on simulated data which is a simple tool for showing some specific traits of a quantity but might still miss some of their essential features affecting the study of real world data.

Thus, we aim to study here the differences between these measures of dependence. We are interested in what kind of properties they have in theory and whether these qualities also work for both real and simulated data sets. We want to compare different coefficients to find which one of them recognises dependence most effectively, in the cases where the relationships between the variables are linear or not, monotonic or not, and functional or not.

The structure of this thesis is as follows. First, in Section 2, we show the fundamental definitions of all the related concepts and introduce the known properties that these quantities should have according to the literature. In Section 3, we then study how these coefficients behave when computed with the programming language R from simulated data about different relationships and, for instance, what kind of

an impact statistical noise and the number of observations have on the obtained results. In Section 4, we investigate a very recent neural network algorithm that can be used to estimate the value of the mutual information. In Section 5, we experiment with real data sets about a few different topics to see if the coefficients behave as in the earlier simulations for this type of data, too.

At the end of this thesis, there is an appendix section containing the most crucial R and PyTorch codes used in this work. More information can be about the topics can found in the works listed in References, especially in [1, 2, 3, 14, 28, 39, 41, 45, 52, 55]. Note that there is also Index on page 62 where the page numbers for the definitions of the central concepts are listed. I have personally made all the figures in this work by using the vector graphics editor Inkscape, the plots from RStudio and the latex package TikZ.

Finally, I would like to thank my supervisors Professor Janne Kujala and Professor Riku Klén for their useful and constructive suggestions, and Professor Matti Vuorinen for his careful proofreading.

2 Fundamentals

Let us next introduce the different concepts used in this paper. First, we discuss correlation and show a few correlation coefficients, then move on to entropy and mutual information and, finally, define the maximal information coefficient. In this section, we focus on the definitions of these concepts, but more details about their behavior and the methods for their computation can be found in later sections.

2.1 Correlation

In the real world, there are numerous variable pairs that might have either positive or negative association between them. For instance, some type of a relation can be observed between the amount of snow in Finland and the number of migratory bird species commonly present, the number of published works by a researcher and the time since their first publication, or the tweeting frequency and the follower count of a Twitter account. In statistics, *correlation* is a simple yet important concept for describing these kinds of relationships.

For a pair of numerical random variables X and Y, their correlation can be defined formally by using the *population correlation coefficient* [4, p. 33]

$$\rho = \frac{\mathrm{E}((X - \mu_X)(Y - \mu_Y))}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}, \tag{2.1}$$

where μ_X is the expected value of the variable X. This concept was pioneered in the late-19th century by the British scientist F. Galton, who needed a way to describe the similarities between an individual and its offspring in his study of heredity [15, p. 186]. While the definition (2.1) is not the original formulation of correlation, it is well-justified because its numerator is the expression of *covariance* [4, p. 33]. Note also that this coefficient is not defined for variables with no variation but studying the correlation in this case would not be interesting.

The population correlation coefficient has several useful features. Trivially, the expression of ρ is symmetric with respect to X and Y. By the common properties of the expected value and variance, it can be shown that the value of ρ belongs to the interval [-1,1]. The independence of the variables X and Y, denoted here as $X \perp Y$, implies that their population correlation ρ is 0, [4, p. 33]. Furthermore, if similar values of the variables often occur together, then they are *positively correlated* and $\rho > 0$. Correspondingly, the *negative correlation* indicated by $\rho < 0$ means that the larger values of X are accompanied by the smaller values of Y and vice versa. The further away the value of ρ is from 0, the greater the positive or negative correlation is [52, p. 3868].

It should be pointed out here that the correlation between X and Y does not mean causality $X \to Y$ or $Y \to X$. Because the cold weather and decreasing amount of light cause certain birds to migrate out of Finland for the winter when there is typically at least some snow, the amount of snow and the number of bird species in the country are negatively correlated, even if the snow itself would not directly affect the bird migration. Studying the correlation between two variables is still useful since information about it can be used to explain the unknown factors affecting these two

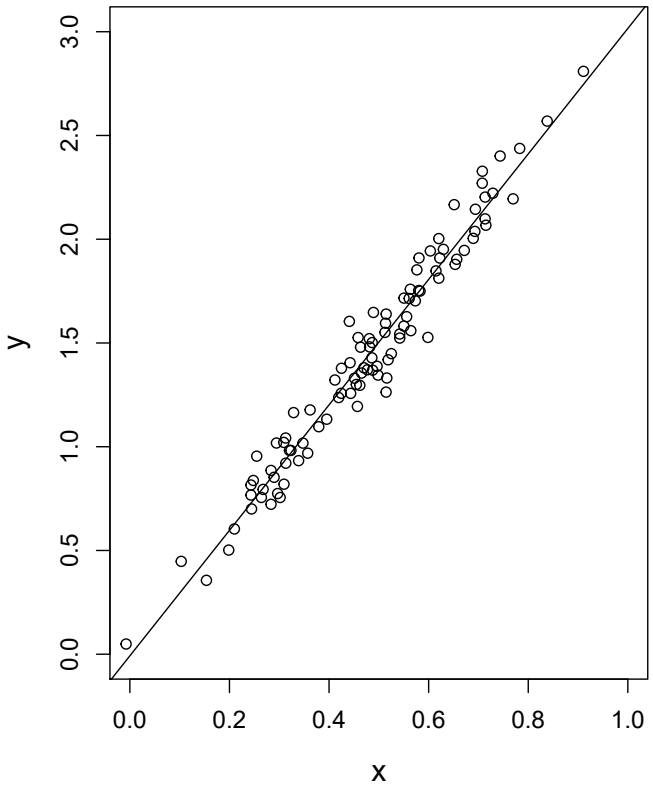


Figure 1: Scatter plot with the least squares regression line, when the data is from two linearly dependent variables.

variables and the values of one variable can be potentially predicted with the other one even if there is no direct causal relationship.

When studying a data set consisting of n pairs (x_i, y_i) of observations with means denoted as $\overline{x}$ and $\overline{y}$, the population correlation can be estimated with *Pearson's correlation coefficient* [52, (4), p. 3868]

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}, \tag{2.2}$$

which was defined in 1895 by the British statistician K. Pearson [26, Table 1, p. 89]. Namely, it follows from the law of large numbers that this coefficient approaches the population correlation coefficient, when n grows large enough. The coefficient r also shares the aforementioned properties of the population correlation coefficient: $r \in [-1, 1]$ by the Cauchy–Schwarz inequality [15, p. 187], $r = 0$ if $X \perp Y$, and the values of r close to 1 express high positive correlation whereas the values approaching -1 mean high negative correlation.

One of the key properties of Pearson's correlation coefficient r is that the slope of the least squares regression line fitted to the scatter plot of the observations (x_i, y_i) can be written as rs_Y/s_X [15, p. 188], where s_X is the *standard deviation* of X

defined as [4, p. 38]

$$s_X = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}.$$

For instance, Figure 1 contains data out of two variables X and Y, for which Pearson's correlation coefficient is $r = 0.979$ and the standard deviations are $s_X = 0.171$ and $s_Y = 0.528$. From these values, we can calculate that the slope of the least squares regression line fitted to the data is just over 3. The value $r = 0.979$ here indicates strong positive correlation between the variables X and Y, which can also be visually verified.

Because of the connection between Pearson's correlation coefficient and the least squares regression, describing the linear relationship with this correlation coefficient feels intuitively very reasonable. Contradictorily, this is also the problem of Pearson's correlation coefficient: While this method is well-suited for studying linear dependence, the coefficient r tells us very little about the underlying relationship if this connection is non-linear. Even if the dependence follows an increasing function such as the cubic function $y = x^3$, Pearson's coefficient can have too low values.

Furthermore, we need to also assume that the marginal distributions of the both variables are normal because otherwise certain outlying observations might affect Pearson's coefficient too much [52, p. 3868]. If we study the correlation between the amount of snow in Finland and some other variable, we must note that there is so little snow on average during a year that the observations collected during a winter blizzard have a very high effect on the value of the coefficient r. While some of the outliers could be simply removed from the data, this will cause information loss.

If the former assumptions about linear dependence and normally distributed variables do not hold, we can measure the correlation in another way: For n pairs (x_i, y_i), *Spearman's correlation coefficient* is [19, (1) & (2), p. 470]

$$r_s = 1 - \frac{\sum_{i=1}^{n}(r(x_i) - r(y_i))^2}{n^3 - n}, \tag{2.3}$$

where $r(x_i)$ is the rank of x_i when the elements in the vector $(x_1, ..., x_n)$ are ordered ascendingly. This coefficient was proposed in 1904 by the British psychologist C. Spearman [52, p. 3866] and, similarly to Pearson's correlation coefficient, its value also varies on the interval $[-1, 1]$, [19, p. 470]. In fact, Spearman's correlation coefficient defined for the pairs (x_i, y_i) is equivalent to Pearson's correlation coefficient for the pairs $(r(x_i), r(y_i))$ of the rank numbers [52, p. 3869].

Spearman's correlation coefficient is often a better choice than Pearson's correlation coefficient if the dependence is monotonic but non-linear. Because Spearman's correlation coefficient is non-parametric, it can also be used for such data where the variables are not normally distributed and no other assumptions about their frequency are needed either. Unlike Pearson's correlation coefficient, Spearman's coefficient can also be used in such situations where one or both of the variables considered is not directly numerical but ordinal, like the level of education, so that integer values can be assigned to its values. Furthermore, while this coefficient was defined above by determining the ranks of the ascending orders of the vectors

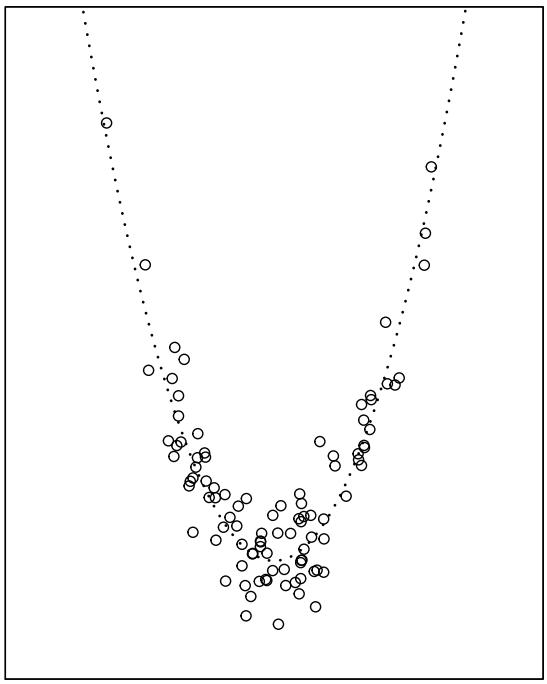


Figure 2: Example of a non-monotonic functional relationship.

$(x_1, ..., x_n)$ and $(y_1, ..., y_n)$, there is nothing preventing us from calculating the coefficient from the descending orders instead if we just order both of these two vectors in the same way. [52, p. 3869]

A third way to estimate the correlation from a data set consisting of n pairs (x_i, y_i) is to use *Kendall's correlation coefficient* [52, (6), p. 3869]

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j) \tag{2.4}$$

introduced in 1948 by another British statistician, M. Kendall [52, (6), p. 3866]. Because the pairs (x_i, y_i) and (x_j, y_j) are *concordant* if $\text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j) = 1$, and *disconcordant* if $\text{sgn}(x_i - x_j)\text{sgn}(y_i - y_j) = -1$, the sum expression above could also be written as the difference between the numbers of concordant and disconcordant pairs in the data. Typically, the values of both Spearman's and Kendall's coefficients are close to each other but the latter one is sometimes less sensitive to error due to it not using squared distances [52, p. 3869].

However, using Spearman's or Kendall's correlation coefficient instead of Pearson's correlation coefficient does not solve all the problems related to correlation. While the dependence in the data does not need to be linear so that it can be represented properly with the coefficient r_s and τ, it should still be monotonic [52, p. 3869]. Namely, if there is neither increasing nor decreasing function describing the dependence, then it is possible that the scatter plot of the data is symmetric in such a way that all the correlation coefficients are 0, even in the case when there would be clear association between the variables considered.

Consider the dependence between the received amount of some drug and the duration of a disease as an example. Suppose that there is an ideal drug dose for the

disease so that both too high and low doses of the drug prolong the disease instead of curing it immediately. Clearly, we can collect such data out of this phenomenon where all three correlation coefficients are 0, even though there is a simple connection between the drug dose and the length of the disease, and therefore none of these coefficient gives any useful information in this case. For instance, the absolute values of all three aforementioned correlation coefficients are less than 0.04 for the data of Figure 2, even though there is a clear parabolic relationship. Thus, yet another alternative method for measuring correlation from non-monotonic data is needed.

For the random variables X and Y, their *maximal correlation coefficient* is [1, (1), p. 27]

$$\rho_{\max} = \sup\{\mathrm{E}(f_0(X)f_1(Y))\}, \tag{2.5}$$

where the supremum is taken over all real-valued functions f_0, f_1 defined in the sets of all the possible values of the variables X and Y, respectively, such that $\mathrm{E}(f_0(X)) = \mathrm{E}(f_1(Y)) = 0$ and $\mathrm{E}(f_0(X)^2) = \mathrm{E}(f_1(Y)^2) = 1$. Originally, this coefficient was proposed in 1941 by H. Gebelein [3, pp. 587-589]. Note that the definition above could be written equivalently as [1, (1), p. 27]

$$\rho_{\max} = \sup\{\rho(f_0(X), f_1(Y))\},$$

when $\rho(X, Y)$ denotes the population correlation coefficient ρ computed for the variables X and Y as in (2.1).

The maximal correlation coefficient fulfills all the requirements that A. Rényi suggested for a measure of dependence in his work in 1959. Namely, this coefficient $\rho_{\max}$ is trivially symmetric with respect to X and Y, and its values vary on the interval $[0, 1]$ so that $\rho_{\max} = 0$ if and only if $X \perp Y$, and $\rho_{\max} = 1$ if and only if $X = h(Y)$ or $Y = h(X)$ for some Borel measurable function h. Furthermore, just like the population correlation coefficient, $\rho_{\max}$ is defined for any random variables X and Y with non-zero variance but cannot be calculated if one of these variables is a constant. [3, p. 589]

By using the maximal correlation coefficient, we can find the dependence between the variables X and Y if it is, for instance, quadratic, cubic or exponential [39, Fig. 2.A, p. 1519]. However, this coefficient has its own issues, too. Finding the suitable functions can be often challenging, there needs to be a high enough number of observations to confirm that the recognised shape in the data is not just incidental, and even this approach does not work very well for all relationship types: In [39, Fig. 2.A, p. 1519], it can be seen that the maximal correlation coefficient is not very effective method when the dependence between X and Y follows a sinusoidal curve, at least according to the results of the article [39]. Furthermore, there are also types of dependence that cannot be described with one function, such as the cross-shaped dependence in Figure 3.

To conclude, the issue with correlation is that while its value is 0 for independent variables, the correlation of 0 does not always mean that the variables are independent. The correlation coefficients typically work well if the dependence is linear or at least monotonic, but other kind of relationships between variables are more difficult to find. Consequently, additional methods to study non-monotonic and non-functional association between two random variables are very much needed.

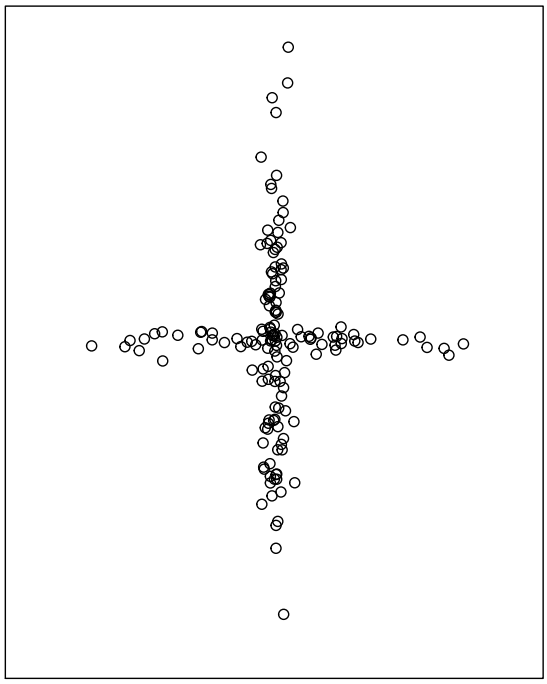


Figure 3: Example of a non-functional relationship in the shape of a cross.

2.2 Entropy

In order to truly understand the connection between two random variables, we need to explain how these variables transmit information. Namely, we must have a way to describe how much information a random variable can give us, or how much information we should expect from it. This question leads us to the concept of entropy, which is one of the most important notions of information theory.

Consider first a singing competition in which each contestant performs multiple times, judges give points directly after the performances and the person with most points in total wins. Suppose that we are only interested in who wins or loses. If a contestant already has so many points that their win is sure, there is nothing surprising about the contest for us anymore. Similarly, witnessing the loss of a contestant that has too few points to win even with perfect final performances would not give us any new information. It is clear that verifying a nearly certain event is not so interesting as observing something highly unlikely.

Consequently, the received amount of information must be decreasing with respect to the probability of an observation. Furthermore, as noted in [42, p. 54], multiple independent events all happening should give us as much information as the sum of each separate event occurring. From these properties, it follows that measuring the amount of information with some logarithm function of the probability of the joint event would be well-founded and, since the amount of information is commonly measured by binary digits called "bits" or natural units, the binary and natural logarithms are good options.

Let us now formally define entropy introduced by the famous mathematician C. Shannon in his ground-breaking work about information theory in the 1940s. For a

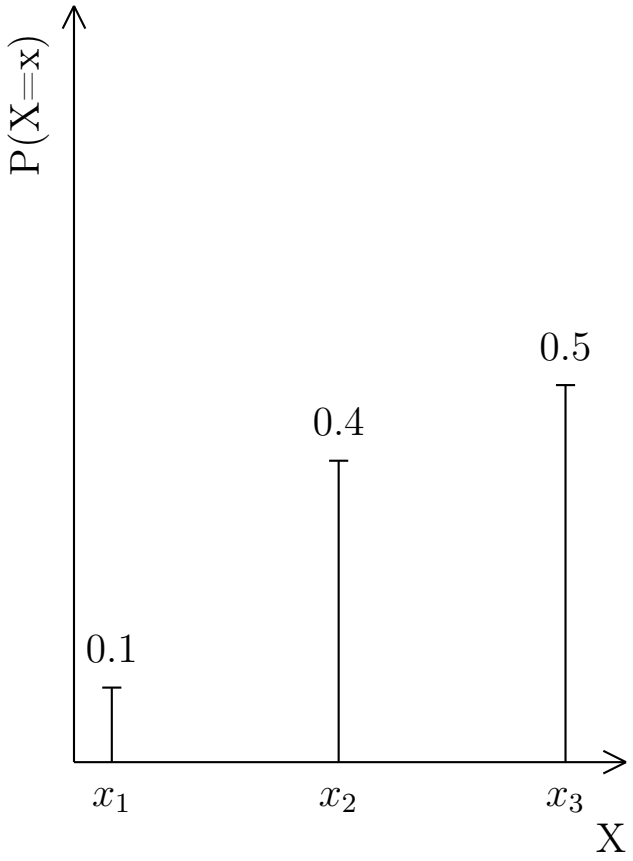


Figure 4: The probability mass function of the discrete random variable X which obtains values x_1, x_2, x_3 with probabilities 0.1, 0.4 and 0.5, respectively.

discrete random variable X with possible values x_i, its *entropy* is [50, p. 430]

$$H(X) = -\sum_i p(x_i) \log(p(x_i))$$

and, for a continuous random variable X with a value set $\mathcal{X}$, let [50, p. 431]

$$H(X) = -\int_{x \in \mathcal{X}} p(x) \log(p(x)) dx.$$

instead. Here, $p(x)$ is the probability of the event $X = x$ or the probability density function of the random variable X, and the base of the logarithm log can be here chosen freely as long as it is over 1, [42, p. 55]. For instance, the entropy of the random variable X of Figure 4 is

$$H(X) = -0.1 \log(0.1) - 0.4 \log(0.4) - 0.5 \log(0.5),$$

which is approximately 1.361 bits or 0.943 natural units. By the definition of the expected value, this definition of entropy is equivalent to the expected value of information given by the random variable X, if its observation x gives an amount of $-\log(p(x))$ of information [42, p. 55].

The definition above can also be extended for the joint and conditional entropy: If the random variables X and Y are continuous, then [44, (2.126), (2.128) & (2.129), p. 100]

$$\begin{aligned} H(X,Y) &= -\int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(x,y)\log(p(x,y))dxdy, \\ H(X|Y) &= -\int_{x\in\mathcal{X}}\int_{y\in\mathcal{Y}} p(x,y)\log(p(x|y))dxdy, \\ H(X|Y=y) &= -\int_{x\in\mathcal{X}} p(x,y)\log(p(x|y))dx \end{aligned}$$

where $p(x,y)$ is the joint probability density function of X and Y, and $p(x|y)$ is the conditional probability density function of X given $Y=y$, see for instance [25, Def. 1.31, p. 11] for the definition. The joint entropy $H(X,Y)$ expresses the uncertainty related to the all possible combinations of the values of the variables X and Y, the first definition for the conditional entropy $H(X|Y)$ tells us the remaining entropy of X once the value of Y is known, and the expression of $H(X|Y=y)$ tells the entropy of X in the case the value of Y is fixed to y. These definitions could also be written for discrete variables by replacing the integral with a sum [50, pp. 430-431].

Note that entropy fulfills the inequality [50, pp. 430-431]

$$0 \leq H(X|Y) \leq H(X) \leq H(X,Y) \leq H(X)+H(Y),$$

where the equalities [50, pp. 430-431]

$$H(X|Y) = H(X) \quad \text{and} \quad H(X,Y) = H(X)+H(Y),$$

hold if $X \perp Y$.

By *Jensen's inequality*, a convex function $f:\mathbb{R}\to\mathbb{R}$ fulfills [34, (1), p. 403]

$$f\left(\sum_{i=n}^{n} p_i x_i\right) \leq \sum_{i=n}^{n} p_i f(x_i), \tag{2.6}$$

where $(x_1,...,x_n)\in\mathbb{R}^n$ and $(p_1,...,p_n)\in(0,1]^n$ such that $\sum_{i=1}^n p_i = 1$, and if f is concave instead, then the reverse of this inequality holds. If $k>0$ and $x\in\mathbb{R}$, then by differentiation

$$\frac{\partial^2}{\partial x^2}\log_k x = \frac{\partial^2}{\partial x^2}\left(\frac{\ln x}{\ln k}\right) = \frac{\partial}{\partial x}\left(\frac{1}{x\ln k}\right) = \frac{-1}{x^2\ln k} < 0,$$

which shows us that the logarithm with any base $k>1$ is a concave function. It follows from this observation and Jensen's inequality (2.6) that, for a discrete random variable X with n possible values x_i, $i=1,...,n$,

$$H(X) = \sum_{i=1}^{n} p(x_i)\log\left(\frac{1}{p(x_i)}\right) \leq \log\left(\sum_{i=n}^{n} p(x_i)\cdot\frac{1}{p(x_i)}\right) = \log n \tag{2.7}$$

where log is any logarithm with a base $k>1$. Clearly, this upper bound is reached if and only if $p(x_i)=1/n$ for all $i=1,...,n$, which makes intuitively sense: The

element of surprise is greatest in the situation where each possible outcome has an equal probability. Note that Jensen's inequality can also be used to find even better upper bounds than (2.7) for entropy, see for instance [34].

From these properties of entropy, we notice that this concept can be seen as one realization of the idea about a measure indicating how much information about a variable conveys through its own observations but this does not still give a very clear picture of what entropy actually is. In physics, this term is related to disorder, disturbance or uncertainty. While Shannon's entropy has a definition similar to what physicists use, the exact connection between these two meanings of entropy is not known. However, calling some sort of "surprisement" or uncertainty of information by the name of a very theoretical quantity expressing the physical state of disorder gives a certain advantage: In order to prove that this name choice for the statistical concept is incorrect, one would first need a definite answer to what the physical entropy really is. In fact, when asked about the name of entropy, Shannon told that another mathematician J. V. Neumann had suggested it for him for this specific reason [42, p. 58].

However, in the context of cryptography, this concept is more clear to understand. Suppose that the random variable X generates the first symbol x_i in a text from some fixed set $\{x_1, ..., x_n\}$ of possible symbols. This symbol is then encrypted by replacing it with a cryptotext symbol y_i, which is determined by certain encryption rules. Clearly, if a person with the cryptotext is trying to figure out the first symbol x_i of the original text, they should first guess the symbol x_j giving the highest probability $p(x_j|y_i)$. Guessing correctly would be most difficult when $p(x_j|y_i) = 1/n$ for all $j = 1, ..., n$ because entropy is at its greatest in this case. Actually, Shannon's entropy $H(X|Y = y_j)$ is a lower bound for the expected number of guesses needed to figure out the correct symbol x_i if the options are tried out in the descending order of probability $p(x_j|y_i)$ [14, p. 796].

Thus, entropy is an interesting concept that can be applied in very many different types of situations. While it can be difficult to understand, this quantity can be used to describe the expected amount of information from a random variable and the uncertainty related to them. It has a clear mathematical definition that has several desirable implications for both continuous and discrete random variables.

2.3 Mutual information

Studying the connection between variables is often inspired by the underlying question if the values of one variable can be used to estimate or predict the values of the other variable. Entropy can be used to measure how much information a random variable gives about itself, but we are interested in the information about one random value that is obtained through the observations of another variable. Thus, we need to introduce a new concept for this purpose.

For a discrete random variables X and Y with values x_i and y_j, respectively, their *mutual information* is [50, p. 431]

$$I(X;Y) = \sum_i \sum_j p(x_i, y_i) \log \left(\frac{p(x_i, y_j)}{p(x_i)p(y_j)} \right) \tag{2.8}$$

and, for a continuous random variables X and Y with value sets $\mathcal{X}$ and $\mathcal{Y}$, this definition is written as [50, p. 431]

$$I(X;Y) = \int_{x\in\mathcal{X}} \int_{y\in\mathcal{Y}} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy.$$

By using the entropy introduced earlier, this definition of mutual information could be simplified to [50, p. 431]

$$I(X;Y) = H(X) - H(X|Y). \tag{2.9}$$

While the groundwork for mutual information was built by Shannon in his study of information theory, this quantity was first proposed in its current form [31, (14), p. 88] in 1957 by E. H. Linfoot [47, p. 1503].

Let us briefly introduce the key properties of mutual information. We see directly from its definitions that mutual information is symmetric with respect to X and Y. For discrete variables X and Y, this symmetry property and the inequality in [23, p. 428] can be used to show that

$$0 \leq I(X;Y) \leq \min\{H(X), H(Y)\},$$

where the lower bound is reached if and only if $X \perp Y$ and the upper bound is obtained if and only if X fully determines the values of Y or vice versa. If X is a discrete random variable with n possible values, it also follows from above and the inequality (2.7) that

$$0 \leq I(X;Y) \leq \log n. \tag{2.10}$$

Note also here that the base of the logarithm above and in the expression of mutual information can again chosen from the open interval $(1, \infty)$, as long as the choice is consistent with the definition of entropy so that the equality (2.9) holds.

Most importantly, mutual information is a measure of dependence that tells us how much information the values of one variable reveal about the values of the other variable [28, p. 3356]. It can detect both linear and non-linear dependence, and even non-monotonic dependence [39, Fig. 2.A, p. 1519]. It is also *self-equitable* [28, p. 3356], which means that, for any deterministic function f, [28, (3), p. 3355]

$$I(X;Y) = I(f(X);Y) \tag{2.11}$$

if the following condition holds: $X \leftrightarrow f(X) \leftrightarrow Y$ forms a *Markov chain* or, equivalently, the conditional probability distribution of Y fulfills $P(Y|f(X),X) = P(Y|f(X))$. Because of these desirable properties, mutual information has different applications in several scientific domains, including information theory, data science and statistics [2, p. 1].

However, this concept has also some problems. While the values of Pearson's, Spearman's and Kendall's correlation coefficients are always on the interval $[-1, 1]$ and the value of the maximal correlation coefficient is from $[0, 1]$, mutual information does not share this property: It can have values over 1, [47, p. 1503] and, if Y is a non-constant deterministic function of a continuous random variable X, $I[X;Y] = \infty$,

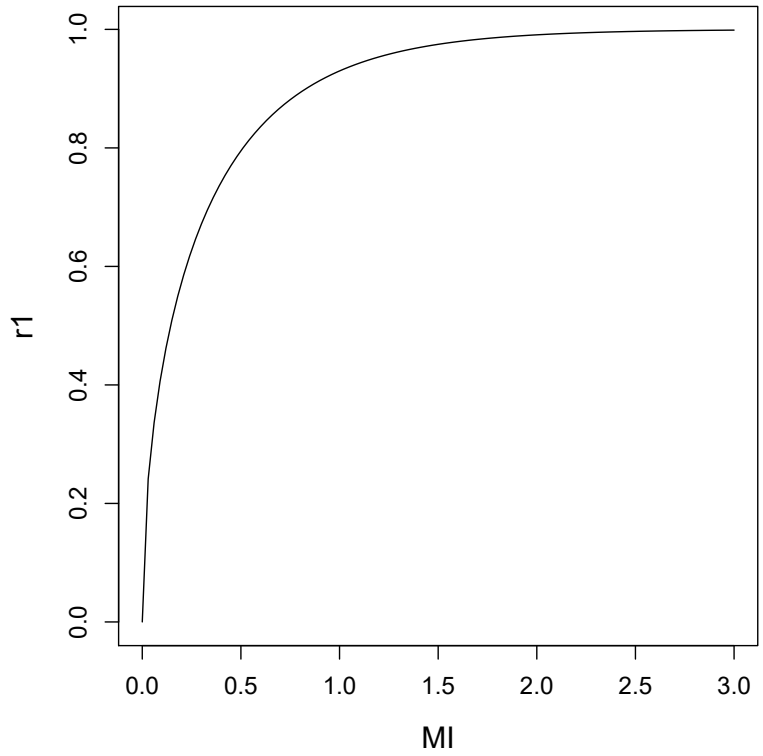


Figure 5: The information coefficient of correlation r_1 as the function of the mutual information (MI).

[28, p. 3356]. This makes the interpretation of the value of mutual information and its comparison to the other ways to measure dependence between variables difficult. For instance, $I(X;Y) = 1.309$ tells us very little about the variables X and Y other than they are not fully independent.

One suggested solution to this issue would be to use the *information coefficient of correlation* [31, (13), p. 88]

$$r_1 = \sqrt{1 - e^{-2 \cdot I(X;Y)}}, \tag{2.12}$$

which was originally proposed also in 1957 by E. H. Linfoot. Like the maximal correlation coefficient, this coefficient obtains values only on the interval from 0 to 1, [31, p. 88]. Clearly, r_1 is strictly increasing with respect to the mutual information so that $r_1 = 0$ for $I(X;Y) = 0$ and $r_1 \to 1^-$ for $I(X;Y) \to \infty$. Figure 5 shows in more detail how the values of this coefficient depend on those of the mutual information.

Another significant issue related to mutual information is that its exact value is difficult to calculate for two continuous variables X and Y [28, p. 3356]. Namely, as we see from the definition of $I(X;Y)$, this would require that we know the distributions of X and Y, which is seldom the case in reality. One possible solution would be to fit some probability distributions to the data and use them to calculate $I(X;Y)$ [28, p. 3356], but there needs to be a lot of data about the random variables X and Y so that one can choose suitable distributions for them.

The mutual information of two variables X and Y can also be estimated from the data by dividing the domain containing all the data points into small intervals

called *bins* and then using the so-called *naive estimate* [28, (6), p. 3356]

$$I_{\text{naive}}(X;Y) = \sum_{\widetilde{x},\widetilde{y}} \hat{p}(\widetilde{x},\widetilde{y}) \log\left(\frac{\hat{p}(\widetilde{x},\widetilde{y})}{\hat{p}(\widetilde{x})\hat{p}(\widetilde{y})}\right), \tag{2.13}$$

where $\hat{p}(\widetilde{x},\widetilde{y})$ is the fraction of data points inside one bin. However, this kind of an estimate systematically gives too large values for the mutual information [28, p. 3356]. Thus, fitting probability distributions for X and Y as suggested above would be a more trustworthy method to compute their mutual information, especially if there is much data [28, p. 3356], but the estimate (2.13) could work as an upper bound for the mutual information.

Mutual information has also one alternative presentation commonly used but, in order to understand it, we need to define one concept first. For two probability distributions P and Q, the *Kuhlback-Leibler divergence* introduced in 1951 by S. Kullback and R. A. Leibler [29] is [2, (4), p. 2]

$$D_{\text{KL}}(P\|Q) = \mathrm{E}_P\left(\log\frac{dP}{dQ}\right), \tag{2.14}$$

where the notation E_P means the expected value taken with respect to the distribution P and dP is the density of the distribution P. For instance, if P and Q are continuous distributions with probability density functions p and q, respectively, then the Kuhlback-Leibler divergence is the integral [17, p. 3802]

$$D_{\text{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(u)\left(\log\frac{p(u)}{q(u)}\right)du.$$

The Kuhlback-Leibler divergence has also the following dual presentation, which was proposed in 1983 by M. Donsker and S. Varadhan: [2, Thm 1, p. 2]

$$D_{\text{KL}}(P\|Q) = \sup_{T:\Omega\to\mathbb{R}} \mathrm{E}_P(T) - \log(\mathrm{E}_Q(e^T)), \tag{2.15}$$

where the supremum is taken over the collection of all such functions T that the two expected values above are finite.

As we see by comparing the definitions of the Kuhlback-Leibler divergence and mutual information, these concepts are clearly connected and, more formally, it holds that [2, (3), p. 2]

$$I(X;Y) = D_{\text{KL}}(P_{XY}\|P_X \otimes P_Y), \tag{2.16}$$

where P_{XY} is the joint distribution of the random variables X and Y and $P_X \otimes P_Y$ is the product of their marginal distributions. This connection is significant because we can use the dual presentation of the Kuhlback-Leibler divergence to compute the value of the mutual information, as pointed out in [2, p. 3]. In fact, using this method together with new neural network algorithms, we can very effectively capture many types of dependence, such as those in the data used to train algorithms to recognize handwritten numbers [2, Fig. 5.c, p. 7].

In conclusion, mutual information is a very functional quantity that can be used to measure how much information the values of one variable convey those of some

different variable, regardless of the exact type of this dependence. Because of this, mutual information works considerably better than the older correlation coefficients when studying diverse relationships. While computation of the exact value of the mutual information has been historically difficult, this problem might be solved with certain newer methods and estimates developed during the recent years.

2.4 Maximal information coefficient

One of the most prominent current changes in the field of statistics is the ever increasing need to find the interesting variables in a data set that might contain several hundreds of them. Since the modern technology has enabled the immense collection of digital data, one must discover the closely related variables more effectively. Consequently, in spite of all the well-defined theoretical properties of mutual information, this over 60 years old concept is alone not enough to fit the requirements of today.

In 2011, D. N. Reshef et al. introduced the *maximal information coefficient (MIC)* [39] defined as [7, p. 2]

$$\mathrm{MIC}(X,Y) = \max_{n_x \times n_y} \frac{\max_G I_G(X,Y)}{\log(\min\{n_x, n_y\})}, \tag{2.17}$$

for the real-valued random variables X and Y, out of which there is some data in the form (x, y). Here, n_x and n_y are the number of bins on the x- and y-axes, G is a $n_x \times n_y$-grid over the plotted data like shown in Figure 6, and $I_G(X,Y)$ is the mutual information under the grid G. In other words, this quantity $I_G(X,Y)$ is computed from the data (x, y) by considering the probability of each box of the grid G proportional to the number of the data points inside the box.

While this method for computing $I_G(X,Y)$ resembles the computation of the naive estimate (2.13), it must be noted that here the aim here is to choose such a grid G that gives the highest possible value for $I_G(X,Y)$. Consequently, the MIC cannot be directly derived from the naive estimate without any sort of maximization. Note also that the product of the numbers n_x and n_y in (2.8) is often limited with some function $B(n)$ depending on the sample size n [12, p. 2]. Furthermore, the logarithm in (2.17) needs to be chosen so that it fits the choice of logarithm in the definition of mutual information (2.8) and the binary logarithm seems to be therefore quite a common option here.

Like mutual information, the MIC is also a tool used to measure dependence that cannot be necessarily found with the simpler correlation coefficients. This is the *generality* property of MIC: The dependence captured by this method is not limited to certain function types such as linear or monotonic, and not even relationships modelled with functions [39, p. 1518]. Furthermore, it follows trivially from the symmetry of mutual information that the MIC is symmetric with respect to X and Y [39, p. 1520].

However, unlike those of mutual information, the values of the MIC only vary on the interval $[0, 1]$, [39, p. 1519]. In fact, this result follows directly from the earlier inequality (2.10), which is enough to show that

$$0 \leq I_G(X,Y) \leq \log(\min\{n_x, n_y\})$$

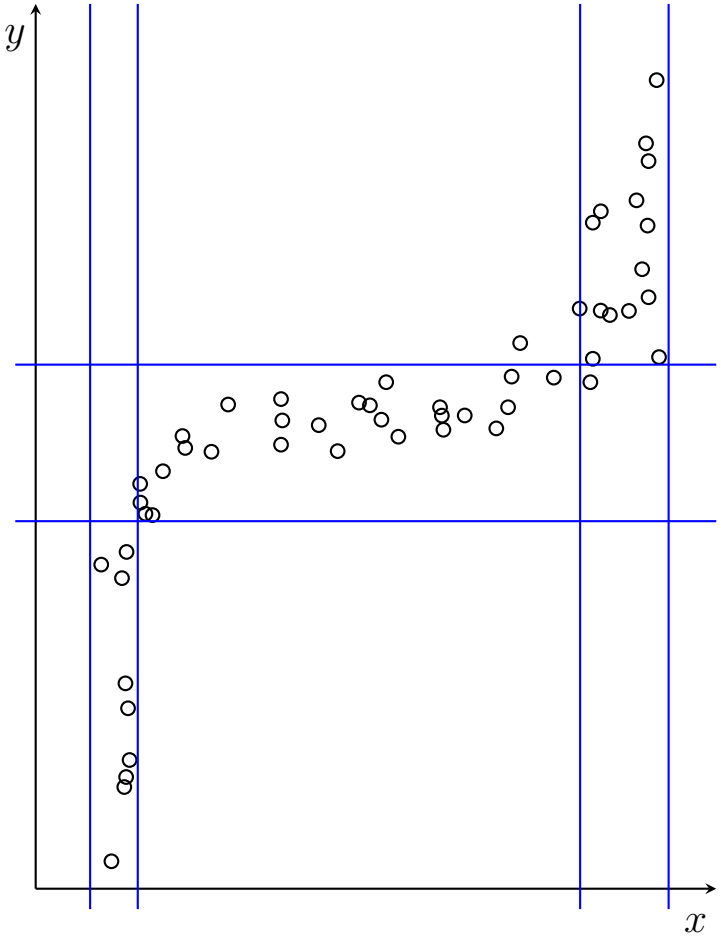


Figure 6: A 3×5-grid over a scatter plot (x_i, y_i) with $n = 53$ observations.

due to the symmetry of mutual information. This property makes the values of the MIC easier to interpret and compare to, for instance, the values of the correlation coefficient.

According to Reshef et al., the MIC also has a similar *equitability* property, which means that the values of the MIC are similar to equally noisy relationships even if the exact type of the dependence varies [39, p. 1518]. Here, by *noisiness*, we mean the numerous unavoidable irregularities in the data which occur because no real life relationship follow some function perfectly. This property resembles the self-equitability of mutual information defined in (2.11), but mutual information itself does not measure the noisiness of different relationships similarly [47, p. 1503].

To compute the noisiness in data, Reshef et al. use the *coefficient of determination* R^2 [39, p. 1518]. While this statistic is typically only defined for linear dependence, see [4, Def. 7.3, p. 265], this definition can be extended: If the function f defines the relationship between two random variables X and Y so that $Y = f(X) + \nu$ with some third variable ν, then for this relationship

$$R^2 = R^2(f(X); Y) = \rho(f(X); Y)^2, \tag{2.18}$$

where $\rho(f(X); Y)$ is the population correlation ρ of the variables $f(X)$ and Y [28, p. 3355]. Consequently, a measure of dependence $D(X; Y)$ is equitable according to

the definition of Reshef et al. if [28, (1), p. 3355]

$$D(X;Y) = g(R^2(f(X);Y)) \tag{2.19}$$

for some function g not depending on the joint probability distribution of the variables X and Y.

One of the main advantages of the MIC is that it is fully non-parametric method. One does not need to know the probability distributions of the variables X and Y to compute the value of $\mathrm{MIC}(X,Y)$, and any errors cannot therefore be made while estimating these distributions, either. However, this does not necessarily mean that the MIC could be computed from lesser amounts of data: Given the definition of MIC, it is clear to see that the more data there is, the more reliable this estimate becomes.

Another interesting side of the MIC is that it is able to recognize multiple trends in the same data [39, Fig. 4, p. 1521]. Suppose for instance that we are interested in the connection between a person's income level and how often they fly. Generally, the number of both the business and holiday flights increases with respect to the income, but some people might avoid flying altogether because of the climate reasons, flight phobia or some motive not related to their financial situation in any way. Because of this, we might obtain a data where these variables plotted against each other follow two distinct curves. While this kind of dependence could not be properly studied with correlation coefficients, the MIC should work in this situation.

One of the issues related to the MIC is that this method is very computationally intensive [7, p. 2]. If there is much data, as there often is in the situations where the MIC is needed, finding the suitable grid G is time-consuming and requires a lot of computation power. One difficulty is calculating the logarithms of proportions required for $I_G(X,Y)$ [47, p. 1503]. Recently, there have been developed a few different algorithms that should work more efficiently for this purpose, see for instance [7, 12, 55], but they give slightly different values for the MIC [7, Table 1, p. 3; Fig. 2, p. 4 & Fig. 3, p. 5] and it is not clear which of these algorithms is the most trustworthy.

Note also that while the value of the MIC should ideally be 0 for independent variables X and Y because mutual information has this property, this does not always work in the reality. Namely, the MIC locates very effectively even the slightest shapes in the data set that could be interpreted as dependence also in the case where there is actually no association between variables. In the numerical results presented by Reshef et al., the value of the MIC for independent random variables was 0.18 while the correlation coefficients and mutual information had all absolute values of 0.03 or less, see [39, Fig. 2.A, p. 1519]. Because of this, one cannot draw direct conclusions that there is dependence between the variables even if the value of the MIC would be, for instance, on the interval from 0.10 to 0.30 or so.

Furthermore, the value of the MIC decreases quickly when the amount of statistical noise increases. In [43], N. Simon and R. Tibshirani criticize the original article introducing the MIC for this very reason. According to their computer simulations, even Pearson's correlation coefficient is a more powerful measure for linear dependence when there is enough noise [43, pp. 1 & 3]. Instead of MIC, Simon and Tibshirani suggested one alternative approach called *distance correlation* [43, p. 1],

which was introduced in 2007 by G. J. Székely et al. [48, Def. 3, p. 2773] but is considerably less studied than mutual information.

However, in a later article [40] in 2013 by Reshef et al., it is pointed out that, while the MIC has low power for detecting weak relationships, it is more equitable than the distance correlation [40, pp. 9-11]. It is important to keep in mind that all these distinct measures of dependence were created for different purposes, so the differences in their behavior are quite expected and do not necessarily mean that one measure would be worse than the others. As mentioned in [40, p. 11], the MIC is a well-suited measure of dependence for situations where we need to compare different relationships and find the strongest one.

Still, while equitability of the MIC would be a preferable property when considering the scientific significance of this quantity, the justification used to prove that the MIC fulfills the criterion of this property is questionable. In the original article [39] by Reshef et al., no mathematical proofs are provided and the analysis is mostly done on simulated data. In fact, in the article [28] from 2014 by J. B. Kinney and G. S. Atwal, it is mathematically proven that no non-trivial measure of dependence can satisfy the definition of equitability (2.19) introduced in 2011 by Reshef et al. and the MIC does not even share the self-equitability property of mutual information. However, in yet another article [41] in 2016 by Reshef et al., the mathematical background of the MIC is explained further and the theoretical properties of MIC are defined in more detail.

There is also one open question related to the MIC about the definition of the MIC between two variables X and Y conditional on some third variable Z. According to T. Speed [47, p. 1503], the definition of the MIC should be extended into the form $\mathrm{MIC}(X, Y|Z)$ to study this case. While formulating the needed expression might be a very straight-forward task, it has not been studied yet how fixing the value of Z affects the ability of the MIC to recognize the dependence between the conditional variables X and Y.

Thus, the MIC is a relatively new non-parametric way to measure dependence between two variables. The values of the MIC might be potentially used to compare the noisiness of distinct relationships, but there is debate in the scientific community about this and also the power of the MIC is known to be low for noisy data. Nonetheless, since the MIC is able to recognize relationships of any type, it might have an important role in several different scientific domains in the future.

3 Simulations with R

In this section, we will study different measures of dependence through simulations written in the programming language R. First, we introduce the functions and packages needed to compute these coefficients and then we build models for several types of relationships between two variables. Finally, we inspect how the values of the measures of dependence change when the amount of statistical noise and the number of observations varies in our models.

3.1 Methods for computation

In each simulation, we compute the values of seven different measures of dependence. These quantities include Pearson's correlation coefficient r defined as in (2.2), Spearman's correlation coefficient r_s (2.3), Kendall's correlation coefficient τ (2.4), the maximal correlation coefficient $\rho_{\max}$ (2.5), mutual information (2.8) and the maximal information coefficient (2.17). Because the value of the mutual information is not directly comparable to the other coefficients, we also calculate the information coefficient of correlation r_1 defined in (2.12).

When writing the code in R, the three correlation coefficients r, r_s and τ can be all computed between two vectors x and y with the same base function called *cor*. We need to just choose correct value of the argument *method* from the options *"pearson"*, *"spearman"* and *"kendall"*. If the method is not otherwise specified, the function returns Pearson's correlation coefficient by default.

For the maximal correlation coefficient $\rho_{\max}$, we need the function *ace* from the R-package *acepack*. Namely, by first transforming the vectors x and y with the function *ace* and then using the aforementioned function *cor* to compute Pearson's coefficient from the output of *ace*, we will attain the coefficient $\rho_{\max}$ of the original vectors x and y. The idea behind this function *ace* is that it uses the alternative conditional expectations algorithm introduced in 1985 L. Breiman and J. H. Friedman [6] to find the suitable transformation needed to maximize the amount of variation in y explained by x [46, p. 2].

The value of the mutual information between two vectors x and y can be computed by using two different functions from the R-package *infotheo*. If we create a dataframe out of the vectors x and y, we can namely use the function *discretize* to discretize the values in this dataframe with equal width binning algorithm and then compute the mutual information with the function *mutinformation* for the discretized data in natural units [37, p. 9]. The information coefficient of correlation r_1 is then computed by applying the formula (2.12) for the mutual information obtained with this method.

Note that the aforementioned functions only give an estimate of the ground truth mutual information. To obtain the exact value, we would need to know the distributions of the random variables X and Y about which our data is collected and compute the mutual information directly from its definition (2.8). The function *discretize* divides data of n observations into $\sqrt[3]{n}$ bins by default [37, p. 4] and, because of this discretization, the returned value of the mutual information is that of the naive estimate (2.13) in these bins. Similarly, the value of r_1 might be slightly

inaccurate, too.

Finally, the value of the MIC is computed with the function *mine* from the R-package *minerva*. This function is the R-version of the older C++, Python and MATLAB functions that are based on the original introduction of the MIC in the article [39] from 2011 by Reshef et al. [20, p. 10]. For two vectors x and y, *mine* computes the MIC from the expression (2.17) where the product $n_x \times n_y$ of the number of bins on the axes is limited by the function [20, p. 9]

$$B(n) = \begin{cases} \max\{n^\alpha, 4\}, & \text{if} \quad \alpha \in (0, 1], \\ \min\{n, \alpha\}, & \text{if} \quad \alpha \geq 4. \end{cases}$$

Here, the parameter α is either determined by an input parameter from the set $(0, 1] \cup \{4, 5, 6, ...\}$ or, if no value is specified by the user, $\alpha = 0.6$ by default. We use this default value in our simulations so that $B(n) = n^{0.6}$, as suggested in [39, p. 1519]. The function *mine* returns a list of five statistics related to maximal information-based non-parametric exploration and the first coefficient in this list is the MIC, named also as *MIC* in the output [20, p. 11].

Out of the three additional packages needed, *acepack* was published in 2016 by P. Spector et al. [46], *infotheo* in 2009 by P. E. Meyer [37] and *minerva* in 2019 by M. Filosi et al. [20]. Each of these three packages should work on the R-version 3.3.0 released on May 3rd, 2016, or newer. In this thesis, all the R-codes were written in RStudio with the R-version 3.4.3, by using the version 1.4.1 of *acepack*, the version 1.2.0 of *infotheo*, and the version 1.5.8 of *minerva*. More details out of these packages and their functions can be found in [46, 37, 20], respectively.

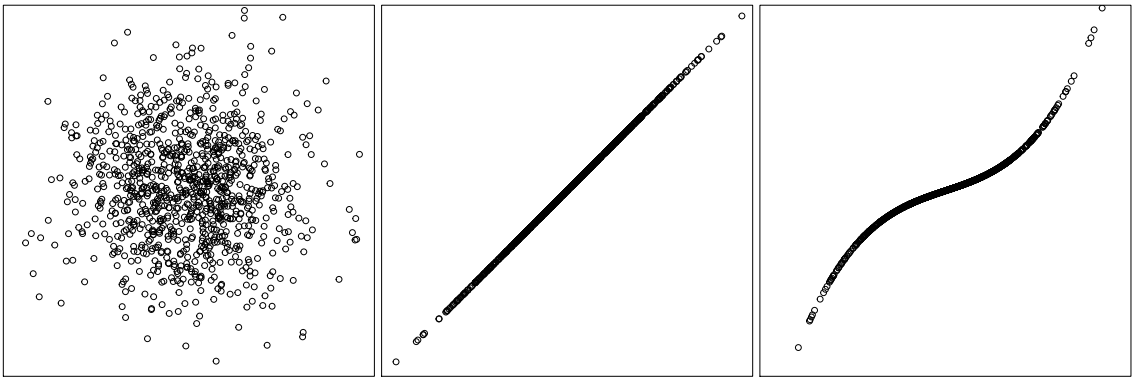
The different measures of dependence could also be studied with several programming languages other than R. The reason why we use R here is that it is simple and works well when generating data from simulations. Because the R-function used for computation of the MIC is very recent, it is also interesting to see how well it performs. Potential issues with using R here are that the returned values for the measures of dependence other than the three correlation coefficients are unlikely to be fully accurate. However, even if there are small differences between the real values of these quantities and their values when computed with the R-functions introduced above, it does not hinder us because our main aim here is provide an overview of the behavior of these coefficients, not find their fully exact values.

3.2 Models

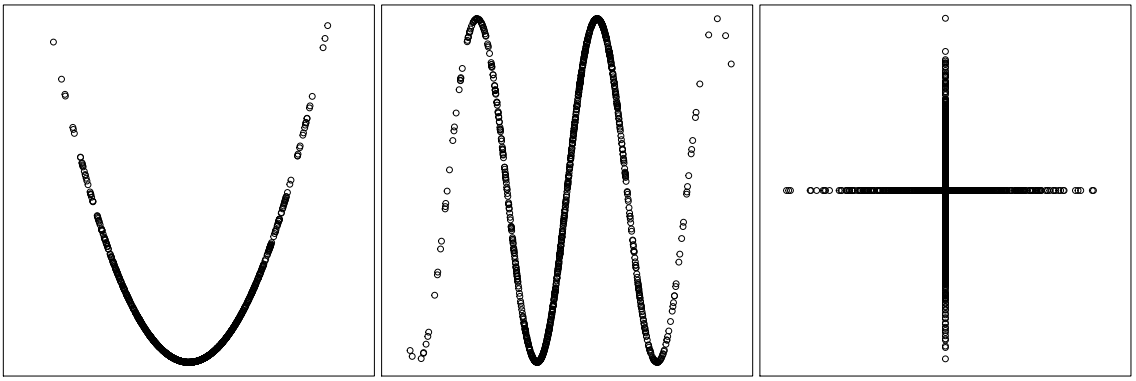
To study the behavior of different measures of dependence, we simulate data sets of n observations (x_i, y_i) according to the following nine models. We build these models here so that $n = 1000$ observations of this model fit very often inside the square $(-1, 1) \times (-1, 1)$ because, in this way, they are more clearly comparable with each other and their noise levels can be adjusted similarly. To illustrate the structure of dependence in the models, Figure 7 contains a scatter plot for a simulation of 1000 observations without any noise from each model.

First, we consider the case of no dependence, where the observations are generated from the model

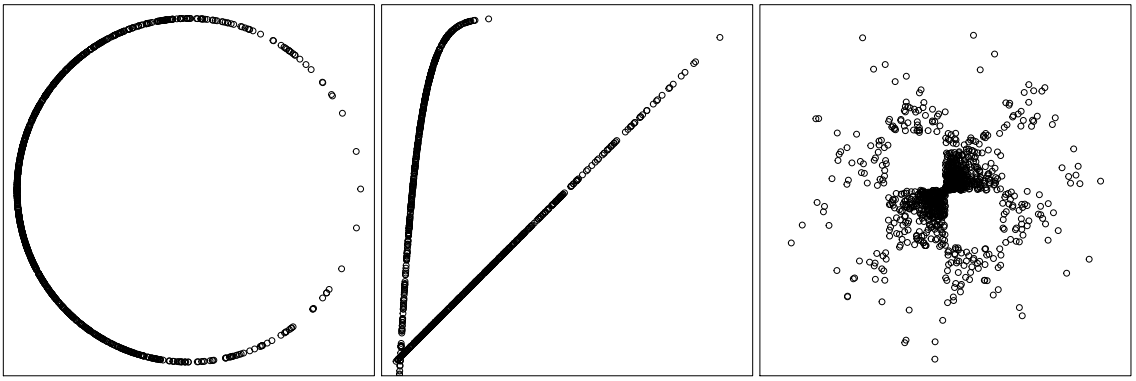
$$x_i \sim N(0, 0.1), \quad y_i \sim N(0, 0.1), \quad i = 1, ..., n. \tag{3.1}$$



(a) No dependence (b) Linear (c) Cubic



(d) Quadratic (e) Sinusoidal (f) Cross



(g) Circular (h) Two functions (i) Checkerboard

Figure 7: Scatter plots of one simulation from the models (3.1)-(3.9) with $\sigma = 0$ and $n = 1000$.

Thus, in other words, our data contains n observations from two independent random variables X and Y following the normal distribution $N(0, 0.1)$. See for the scatter plot of one simulation of this model with $n = 1000$ observations in Figure 7a. Ideally, all the coefficients measuring dependence should have values of 0 for data of this model because $X \perp Y$ here. Furthermore, note that to generate observations from the normal distribution with variance 0.1, one must choose the value $\sqrt{0.1}$ for the third input parameter of the R-function *rnorm* because it is the standard deviation of this distribution.

Next, we consider the case of linear dependence plotted in Figure 7b. Our model is the linear model

$$x_i \sim N(0, 0.1), \quad y_i = x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n, \tag{3.2}$$

where the amount of noise is determined by varying the value of the standard deviation $\sigma \geq 0$. If $\sigma = 0$, then there is no statistical noise and the values of all our coefficients except the mutual information should be 1 in this case.

Our third type of dependence is the cubic dependence

$$x_i \sim N(0, 0.1), \quad y_i = x_i^3 + \frac{1}{3}x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n. \tag{3.3}$$

Because this dependence is monotonic but non-linear, Pearson's correlation coefficient r might not properly recognize it but all the other coefficients should. See Figure 7c for the scatter plot of this model.

The fourth case considered is the quadratic or parabolic dependence of Figure 7d. The model used is now

$$x_i \sim N(0, 0.1), \quad y_i = 3x_i^2 - 1 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n. \tag{3.4}$$

Even though this is a very simple type of dependence, it is non-monotonic. Since the shape of the generated data should be quite symmetric, the values of correlation coefficients r, r_s and τ are likely to be around 0. Because these three coefficients are designed for monotonic dependence, their values do not give any useful information about this dependence or the following other five non-monotonic types of dependence. However, if there is little to no noise in this model, then the values of the maximal correlation $\rho_{\max}$, the information coefficient of correlation r_1 and the MIC should be close to 1.

Our final dependence that can be described with a single-variable function is the sinusoidal dependence, like in Figure 7e. The model here is

$$x_i \sim N(0, 0.1), \quad y_i = \sin(9x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n. \tag{3.5}$$

Note that we use here the coefficient 9 inside the sine function to ensure that there are more than one sinusoid in the final data. All our quantities other than the first three correlation coefficients should technically recognize this type of dependence but, according to the simulations summarized in [39, Fig. 2.A, p. 1519], the MIC might give the greatest values.

The sixth type of dependence is a cross-shaped dependence between the variables X and Y, see Figure 7f. The observations are now generated according to the model

$$\begin{aligned} &k \sim \mathrm{Bin}(n, 0.5), \\ &x_i \sim N(0, \frac{\sigma^2}{4}), \quad y_i \sim N(0, 0.1) \quad \text{for} \quad i = 0, 1, ..., k, \\ &x_i \sim N(0, 0.1), \quad y_i \sim N(0, \frac{\sigma^2}{4}) \quad \text{for} \quad i = k+1, ..., n. \end{aligned} \tag{3.6}$$

Here, we first choose a value from the binomial distribution to determine how many of our observations (x_i, y_i) are on the vertical line segment of the cross and then generate observations. Note that the noise is produced by the normal distribution with variance of $\sigma^2/4$ because changes in the parameter σ cause otherwise too much noise. Still, we only consider values of σ clearly under $2\sqrt{0.1}$ because this model only produces a data of independent observations like the model (3.1) for $\sigma = 2\sqrt{0.1}$. Nonetheless, considering this model is interesting because it is a very simple example of symmetric but non-functional dependence.

Next, we create a circular data like in Figure 7g. In each trial, we choose n observations (x_i, y_i) from the model

$$k_i \sim N(0,1), \quad l_i \sim N(1, \sigma^2), \quad x_i = -l_i \cos(k_i), \quad y_i = -l_i \sin(k_i), \quad i = 1, ..., n, \tag{3.7}$$

where the arguments of the trigonometric functions are radians. In other words, we choose points (x_i, y_i) by using one variable $K \sim N(0, 1)$ determining their angle magnitudes from the negative x-axis in radians and another variable $L \sim N(1, \sigma^2)$ to choose their distance from the origin. If $\sigma = 0$, each observation is a point from the unit circle and, by increasing σ, the amount of noise in this data grows.

One of the properties of the newer measures of dependence is that they should recognize dependence following several distinct functions, so let us test this, too, with one simulation. We use the model

$$\begin{aligned} &k_i \sim N(0,1), \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n; \\ &x_i = \frac{2}{3} k_i - 1, \quad y_i = x_i + \epsilon_i \quad \text{if} \quad k_i \geq 0, \\ &x_i = -(-k_i)^{0.1} + 0.1, \quad y_i = (x_i - 0.1)^{10} + 1 + \epsilon_i \quad \text{if} \quad k_i < 0, \end{aligned} \tag{3.8}$$

to generate the needed observations (x_i, y_i). Consequently, our data is combination of simple linear dependence $y = x$ and the polynomial dependence $y = (x-0.1)^{10}+1$, see Figure 7h.

Our final type of dependency is checkerboard-shaped, which is depicted in Figure 7i. The model is as follows:

$$\begin{aligned} &x_i = k_{i0}, \quad y_i = k_{i1} + \epsilon_i, \quad \epsilon_i \sim N(0, \frac{\sigma^2}{9}), \quad i = 1, ..., n, \quad \text{where} \\ &\begin{pmatrix} k_{i0} \\ k_{i1} \end{pmatrix} \in \left\{ \begin{pmatrix} k_{i0} \\ k_{i1} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right) \;\middle|\; \lfloor 3k_{i0} \rfloor - \lfloor 3k_{i1} \rfloor \equiv 0 \pmod 2 \right\}. \end{aligned} \tag{3.9}$$

Here, $\lfloor x \rfloor$ denotes the floor function that gives the greatest integer less than or equal to the input x. Consequently, we generate paired values (k_0, k_1) of the random variables $K_0 \sim N(0, 0.1)$ and $K_1 \sim N(0, 0.1)$, accept these values if the difference $\lfloor 3k_0 \rfloor - \lfloor 3k_1 \rfloor$ is an even number, and then turn the accepted pairs into the observations (x_i, y_i) by just adding some noise from $N(0, \sigma^2/9)$. If there is no noise, the observations fit inside a few different tiles with side length of 1/3.

3.3 Simulations without noise

Now, we compute the values of Pearson's, Spearman's and Kendall's correlation coefficients, the maximal correlation coefficient $\rho_{\max}$, the mutual information, the informal coefficient of correlation r_1 and the MIC for the data sets simulated from the preceding models (3.1)-(3.9). In order to find realistic values for our quantities and avoid possible bias, we calculate the mean values out of 1000 data sets with $n = 1000$ observations generated from each model. Furthermore, we also set $\sigma = 0$ here so there is no noise in statistical noise and the data sets are as in Figure 7.

Model	r	r_s	τ	$\rho_{\max}$	MI	r_1	MIC
No dependence	$-7.2 \cdot 10^{-4}$	$-6.7 \cdot 10^{-4}$	$-4.6 \cdot 10^{-4}$	0.079	0.033	0.250	0.133
Linear	1.000	1.000	1.000	1.000	2.197	0.994	1.000
Cubic	0.934	1.000	1.000	0.998	2.197	0.994	1.000
Quadratic	$-1.9 \cdot 10^{-3}$	$-2.1 \cdot 10^{-3}$	$-2.1 \cdot 10^{-3}$	1.000	1.404	0.969	1.000
Sinusoidal	0.070	0.162	0.128	0.990	0.991	0.928	1.000
Cross	$-8.4 \cdot 10^{-6}$	$-1.4 \cdot 10^{-5}$	$-1.1 \cdot 10^{-5}$	0.928	0.321	0.688	0.579
Circular	$-9.1 \cdot 10^{-5}$	$6.6 \cdot 10^{-4}$	$8.0 \cdot 10^{-4}$	0.995	1.252	0.958	0.996
Two functions	0.203	0.297	0.356	0.950	1.161	0.949	0.760
Checkerboard	0.058	0.227	0.187	0.877	0.403	0.744	0.573

Table 1: The mean values of Pearson's, Spearman's and Kendall's correlation coefficients (r, r_s, τ), the maximal correlation coefficient $\rho_{\max}$, the mutual information (MI), the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the models (3.1)-(3.9), when $n = 1000$ and $\sigma = 0$.

From Table 1, we see that these coefficients work in a quite expected way in these kinds of simulations. The three correlation coefficients r, r_s and τ all recognize the linear dependence of the model (3.2), but only the two latter coefficients give the value 1 for the non-linear but monotonic cubic dependence of the model (3.3). This result is commonly used to explain the differences between these coefficients: For instance, in [52, Fig. 1, p. 3869], it is also noted that the value of Spearman's correlation coefficient r_s is 1 but Pearson's coefficient r is less than 0.9 for data resembling the noiseless cubic dependence of Figure 7c. Still, Pearson's coefficient has quite a large value because there is clearly positive correlation in the cubic data.

We also notice that the first three correlation coefficients are on average closer to 0 than the other quantities considered for the model (3.1) where there is no dependence. However, it must be taken into account here that the mean values of such coefficients that can obtain values both above and below 0 are not directly comparable to other measures of dependence varying only on the interval [0,1]. The

means of the absolute values of the coefficients r, r_s and τ for the model (3.1) are 0.025, 0.026 and 0.017, respectively, and these values are slightly larger but still closer to 0 than the means of the other coefficients in Table 1.

The mean values of the MIC are 1 for the linear, cubic, quadratic and sinusoidal types of dependence of the models (3.2)-(3.5), which supports the idea that the MIC is general in the way that it captures relationships regardless of their exact function or type. It must be noted here that this result might be partially due to the computational methods, because the function *mine* used to calculate the MIC seem to return quite often the exact value of 1 and give values with only two decimals or less. Furthermore, even though the generality property described in [39, p. 1518] should work also for non-functional relationships, our results do not support this: The mean value of the MIC is less than 0.6 for the cross dependence of the model (3.6), even though it is a very clear and simple type of dependence. The mean of the MIC is also less than 0.6 for the checkerboard dependence of the model (3.9) and less than 0.8 for the dependence built with two functions of the model (3.8).

The results of Table 1 also show the issues in the interpretation of mutual information. We notice that the means of the mutual information are over 2 for the two monotonic relationships but less than 1.5 in all the other cases. When transforming these values to those of the informal coefficient of correlation r_1, most of them attain values larger than 0.9, which would be interpreted as very strong relationship if this result were the value of some correlation coefficient. However, we also notice that, even though the mean value of the mutual information is only slightly over 0.03 for the model (3.1) with no dependence, the mean of r_1 is 0.25, which is much greater than the corresponding means of the correlation coefficients and the MIC.

According to these simulations, the maximal correlation coefficient works the best if we are looking for a coefficient that recognizes dependence. Namely, the value of $\rho_{\max}$ is relatively small in the case of no dependence, but it is still very close to 1 in all other cases. Even when the dependence is in the shape of a cross or a checkerboard, the maximal correlation coefficient finds it considerably better than the rest of the quantities considered: The mean of $\rho_{\max}$ is 0.93 for the cross model (3.6) and 0.88 for the checkerboard model (3.9).

However, the types of dependence of Figure 7 and Table 1 are not realistic. While they are possible in simulations, there is no relationship in the real world that would follow some function or other theoretical model perfectly without any statistical noise. Consequently, it is important to also study what kind of an impact noise has on the values of the different measures of dependence in these simulations, as we will investigate next.

3.4 Effect of noise

In this subsection, we study how the amount of statistical noise affects the values of the different measures of dependence. Just like earlier, we mostly consider the mean values of these different quantities in 1000 data sets simulated from our models. The number n of observations is fixed here to 1000, too, but we vary the value of the parameter σ in the models to alternate the noise levels in the data sets.

Suppose that we are interested in finding such quantity that gives the value

1 in case of dependence without any statistical noise, then decreases with respect to the amount of noise and has a value of 0 if there is no dependence between variables. However, this leads to the question whether the values close to 0 should be interpreted as independent variables or a very noisy relationship. For instance, in 1000 simulations with $n = 100$ observations from the model (3.1), MIC had some values over 0.35 even though there is no dependence and its values should therefore be clearly over this for noisy but still recognisable relationships so that we could draw any conclusions.

Consequently, it is useful to know how our measures of dependence decrease compared to each other when the amount of noise grows. It can be visually checked that the relationships of the models (3.2)-(3.6) and (3.8) are still discernible if $n = 1000$ and $\sigma = 0.3$. Thus, we study here especially how much our quantities diminish for the interval $[0, 0.3]$ of the parameter σ. Furthermore, note that we discuss here the actual values of these measures of dependence and, unlike in [35, 41, 43], do not focus on their *power*, which means the probability of rejecting a false hypothesis in a statistical test or, in this case, distinguishing the cases of no dependence from the others based on the values of some specific coefficient.

We first study the case of linear dependence. We consider here the six measures of dependence including all other coefficients of Table 1 except the mutual information because it is not comparable to the others and the values of the informal coefficient of correlation r_1 can be used to obtain the mutual information if necessary. We compute the mean values of these six coefficients in 1000 simulations generated from the model (3.2) with $n = 1000$ for each value of $\sigma = 0, 0.03, ..., 0.3$ and draw Figure 8 using this data.

From Figure 8, we see that the mean values of each coefficient decrease as the amount of noise increases. Interestingly, we notice that for all values of σ, the three correlation coefficients fulfill the inequality $r \geq r_s \geq \tau$. While Pearson's and Spearman's coefficients are quite close to each other, the distance from them to Kendall's coefficient τ increases considerably as the parameter σ grows.

Furthermore, we also see that the means of Pearson's correlation coefficient and the maximal correlation coefficient are nearly the same, as even their values cannot be properly distinguished from each other in Figure 8. However, this is not very surprising when we recall how the value of $\rho_{\max}$ is computed here: We transform the vectors x and y so that the proportion of y explained by x is at maximum and then calculate Pearson's coefficient from the transformed vectors. Since the proportion of y accounted by x is already at greatest in the linear case, no transformation is necessary and these two coefficients have equivalent values.

As we see from Figure 8, the values of the MIC decrease very quickly compared to the other quantities. This observation agrees with the earlier simulation results presented by N. Simon and R. Tibshirani in [43, p. 3], and by A. Luedtke and L. Tran in [35, Fig. 5, p. 14]: Even Pearson's correlation coefficient detects linear dependence better than the MIC if there is at least some statistical noise in the simulation. In fact, if the value of σ exceeds 0.1 or so, the means of the MIC are lower than those of all the other coefficients considered.

Let us now consider the impact of noise on the cubic model (3.3). Figure 9 contains the mean values of the six different measures of dependence out of 1000

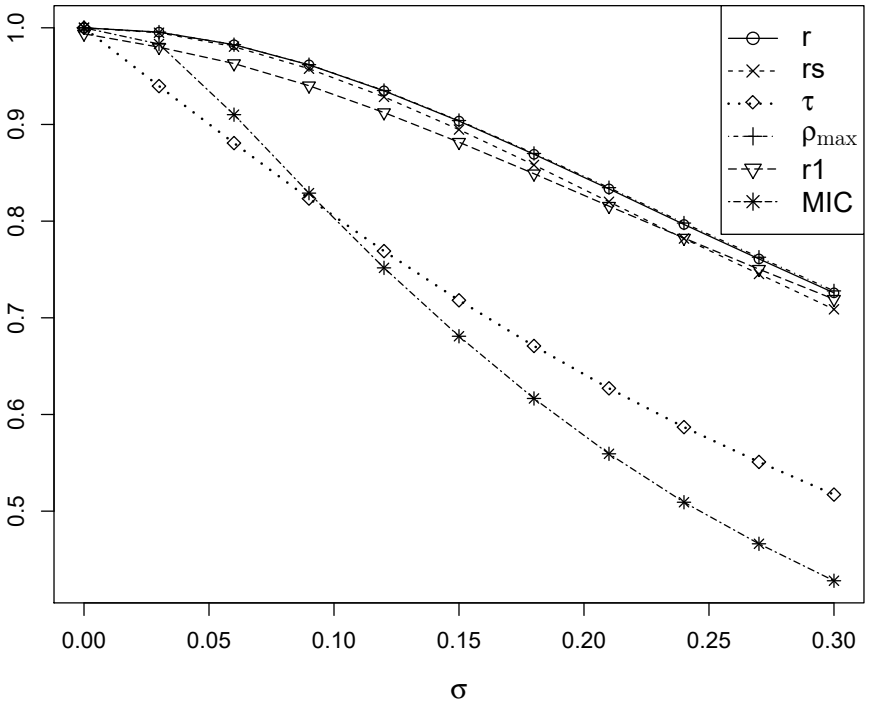


Figure 8: The mean values of Pearson's, Spearman's and Kendall's correlation coefficients (r, r_s, τ), the maximal correlation coefficient $\rho_{\max}$, the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the linear model (3.2) for $n = 1000$ and $\sigma = 0, 0.03, ..., 0.3$.

simulations with $n = 1000$ observations from this model for $\sigma = 0, 0.03, ..., 0.3$. In other words, Figure 9 is plotted in the exactly same way for the cubic dependence as Figure 9 is for the linear dependence.

When $\sigma = 0$, we see from Figure 9 that the mean values of each coefficient are close to 1 but Pearson's correlation coefficient is noticeably lower than the others. This explanation for this is the same one as mentioned earlier when studying the values of Table 1: Pearson's coefficient does not recognize the cubic dependence as well as some other coefficients do because of the non-linearity but, since this function $y = x^3 + x/3$ used in the model (3.3) is strictly increasing, Pearson's coefficient still detects strong positive correlation.

As the value of σ grows, we see that there is clearly more variation between the means of the different coefficient in the case of the cubic dependence than for the linear data and, for instance, the values of Pearson's correlation coefficient and the maximal correlation coefficient are not so close to each other in Figure 9. In fact, for $\sigma > 0.1$, the mean values of the six coefficients fulfill clearly the inequality

$$\text{MIC} < \tau < r_s < r_1 < r < \rho_{\max}.$$

It is also noteworthy that the mean values of the MIC and Kendall's coefficient τ are considerably lower than the others when $0.03 < \sigma < 0.3$. Recall that Kendall's correlation coefficient should be less susceptible to the potential errors because of the absolute distances in its definition (2.4) [52, p. 3869], because this might also explain why it has lower values than other correlation coefficients.

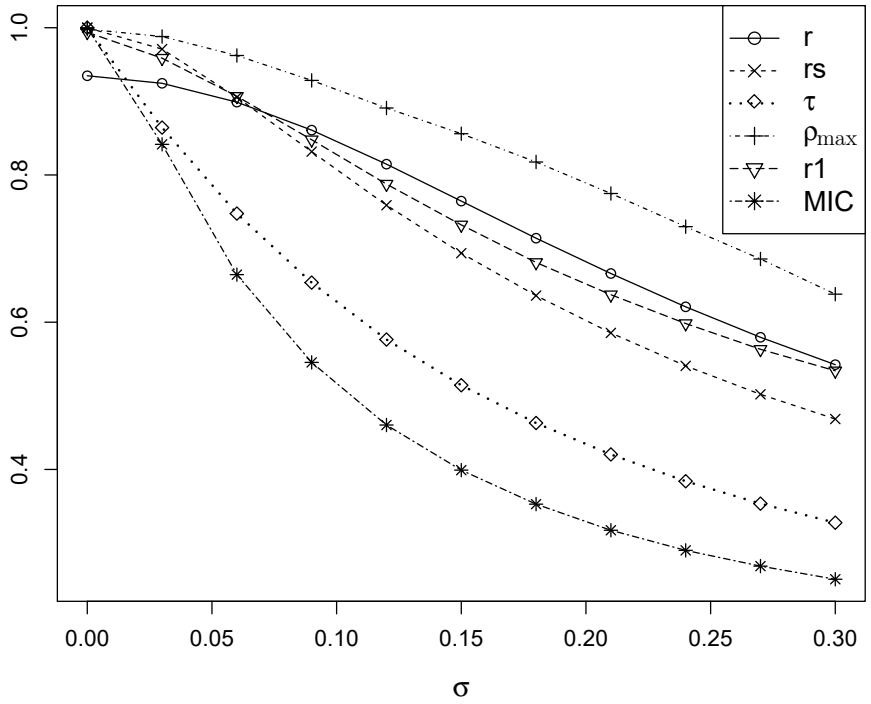


Figure 9: The mean values of Pearson's, Spearman's and Kendall's correlation coefficients (r, r_s, τ), the maximal correlation coefficient $\rho_{\max}$, the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the cubic model (3.3) for $n = 1000$ and $\sigma = 0, 0.03, ..., 0.3$.

The interesting aspect in our simulation is that even though Pearson's coefficient is not designed for non-linear dependence, it has values larger than the MIC that should work for nearly every type of dependence. This result is not new: Also in Simon and Tibshirani's simulations about cubic dependence, Pearson's coefficient r works better than the MIC when there is enough noise [43, p. 3]. According to our simulation here, the maximal correlation coefficient $\rho_{\max}$ seems to be here the best choice if we are looking for a coefficient that detects this kind of dependence and still gives relatively good results even when the amount of noise grows.

Next, let us yet study the case of the two-function dependence that follows the model (3.8). Namely, as can be seen from Table 1, at least the maximal correlation coefficient $\rho_{\max}$ and the information coefficient of correlation r_1 detect this dependence very well and, since it is not symmetric, the values of the first three correlation coefficients clearly differ from 0. Figure 10 contains the means of the six measures of dependence out of 1000 simulations for the model (3.8) with $n = 1000$ and $\sigma = 0, 0.03, ..., 0.3$.

From Figure 10, we can very easily notice which of our quantities can find non-functional dependence: The mean values of three correlation coefficients r, r_s and τ are all less than 0.37, while the means of the three other coefficients stay mostly over this. However, the means of the MIC are still considerably less than those of $\rho_{\max}$ and r_1, and they actually decrease from 0.76 to less than 0.4. Furthermore, Pearson's and Spearman's correlation coefficients stay relatively the same while the mean values of all the other coefficients lessen considerably.

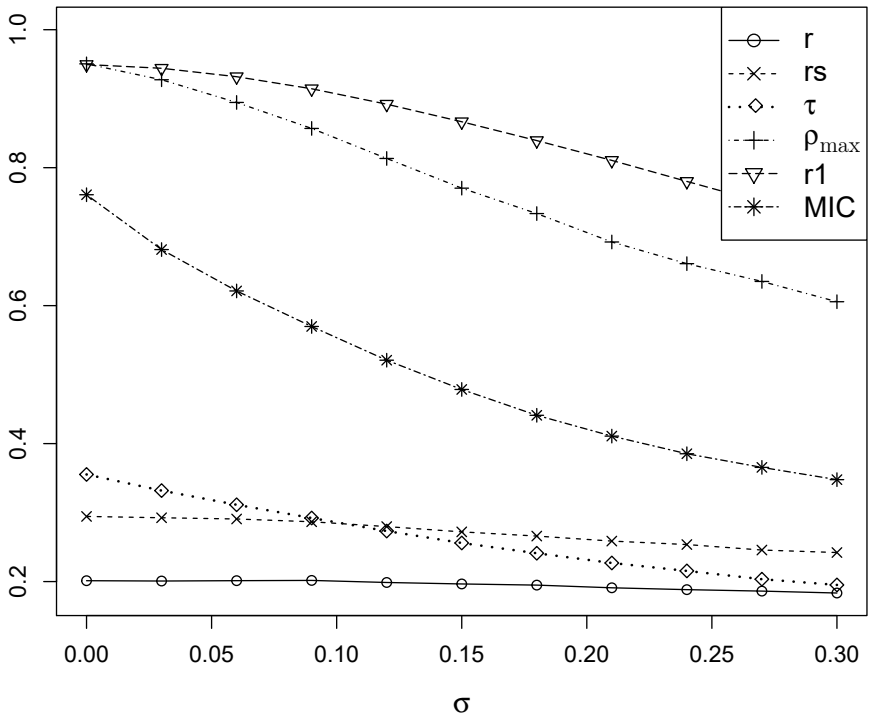


Figure 10: The mean values of Pearson's, Spearman's and Kendall's correlation coefficients (r, r_s, τ), the maximal correlation coefficient $\rho_{\max}$, the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the two-function model (3.8) for $n = 1000$ and $\sigma = 0, 0.03, ..., 0.3$.

We also notice that, in Figures 10, the means of the information coefficient of correlation r_1 are greater than those of $\rho_{\max}$ here, unlike in Figures 8 and 9. Since the dependence cannot be properly described with just one function, it makes more sense that the coefficient r_1 recognizes it better. Recall also that the value of $\rho_{\max}$ is computed by transforming the data points to maximize their correlation but, when the data follows two functions, this maximization cannot be properly done with just one transformation. However, the difference between the means of the MIC and $\rho_{\max}$ is quite unexpected because, according to [39, Fig. 4, p. 1521], the MIC should detect two-function relationships well.

Next, let us yet consider the sinusoidal dependence of the model (3.5). This is interesting because the sinusoidal dependence is clearly non-monotonic unlike the linear and cubic dependence and, according to Simon and Tibshirani's simulations in [43, p. 3], the MIC should recognize it quite well, at least if the period of the sine function used is small enough. Since the first three correlation coefficients do not work for non-monotonic relationships and would just be close to 0, we form here a smaller table with only the means of the three other quantities for $\sigma = 0, 0.03, 0.1, 0.15, 0.3, 0.5, 1$.

From Table 2, we see that the mean values of the maximal correlation coefficient $\rho_{\max}$ exceed those of the MIC around $\sigma = 0.1$ and the means of the informal coefficient of correlation r_1 surpasses MIC, too, when σ is around 0.3. The maximal correlation coefficient has clearly the greatest mean values here if not counting the first few means of MIC. Consequently, most of our tests so far have suggested that

σ	$\rho_{\max}$	r_1	MIC
0	0.990	0.929	1.000
0.03	0.989	0.925	1.000
0.1	0.982	0.908	0.988
0.15	0.974	0.892	0.956
0.3	0.928	0.838	0.835
0.5	0.825	0.753	0.626
1	0.549	0.562	0.317

Table 2: The mean values of the maximal correlation coefficient $\rho_{\max}$, the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the sinusoidal model (3.5), when $n = 1000$ and σ varies.

studying the value of $\rho_{\max}$ would be the most effective way to determine if there is dependence in the data or not.

One curious observation from Table 2 is that the impact of the amount of noise has on the different coefficients is considerably less than it is for the previously considered dependence types. Namely, the means of $\rho_{\max}$, r_1 and the MIC decrease less with respect to σ for the sinusoidal dependence (3.5) than for the models (3.2), (3.3) and (3.8) and, for instance, the MIC is over 0.8 for $\sigma = 0.3$ in Table 2, unlike in Figures 8-10. This is easy to explain: If we have a model

$$y = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n, \tag{3.10}$$

the amount of noise on the plot (x, y) depends on the slope of the function f because the steeper the slope around the point x_i is, the closer the point $y_i = f(x_i) + \epsilon_i$ is to some other point on the function f. Since the sine function with short period has several nearly vertical parts, see Figure 7e, increasing the parameter σ does not affect the noisiness of this model so much.

Consequently, to study the values of our measures of dependence in equally noisy data sets generated from distinct models, we cannot just consider the value of σ. In order to research the equitability of the MIC, recall the definition of the coefficient of determination R^2 from (2.18). Since its value can be easily computed from Pearson's correlation coefficient found with the R-function *cor*, we can compare the connection between the values of the MIC against the noise levels of such models that can be written as in (3.10) for some function f.

Figure 11 depicts the values of the MIC against the statistics R^2 in 1000 different simulations of the linear, cubic, quadratic and sinusoidal models (3.2)-(3.5) with 1000 observations. Note that the level of noise is decreasing with respect to R^2, so the values of the MIC lessen here as the amount of noise increases, just like in Figures 8-10 and Table 2. Alternatively, we could also have plotted the values of the MIC against the quantity $1 - R^2$ that directly describes the amount of noise, but the information given by this kind of a figure would have been the same as that of Figure 11.

As we can see from Figure 11, the values of the MIC in the linear model are less than those in the sinusoidal model but higher than in the values of the cubic and quadratic models, especially if $R^2 > 0.4$. Because of these differences, Figure

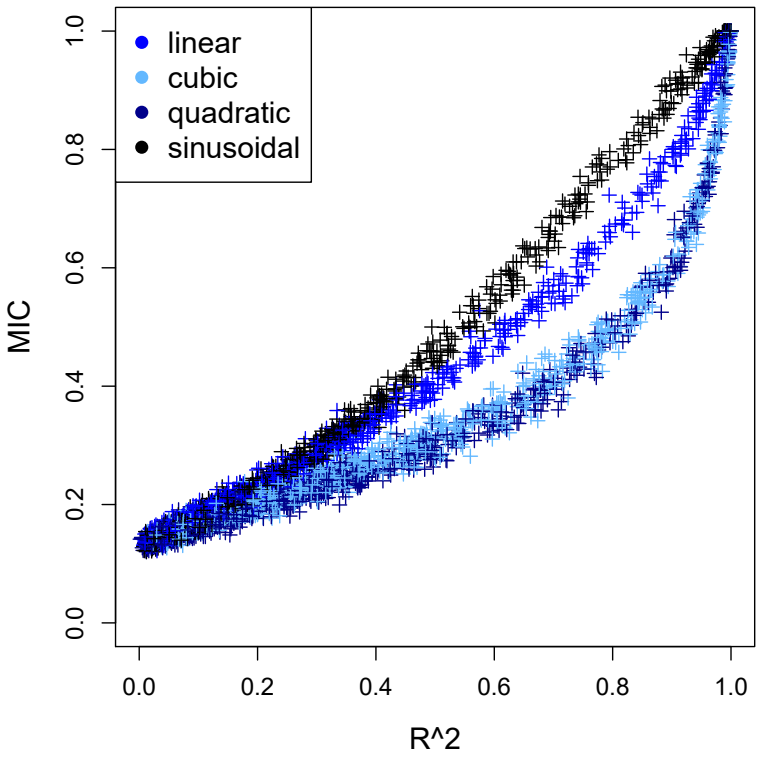


Figure 11: The values of the MIC and R^2 in 1000 simulations of the linear, cubic, quadratic and sinusoidal models (3.2)-(3.5) with $n = 1000$.

11 differs slightly from the earlier results by Reshef et al. in [39, Fig. 2.B, p. 1519], but it must be noted that the computational method used to find the MIC here is not necessarily fully exact. Furthermore, the values of the MIC in both the cubic and quadratic model stay close to each other for all the values of R^2, which supports the assumption that the MIC has some kind of an equitability property against the noise levels measured with the statistics R^2.

To conclude, the maximal correlation coefficient $\rho_{\max}$ is quite a good tool for detecting different types of dependence and its values stay relatively high even when the amount of noise grows. Also, the information coefficient of correlation r_1 works for this purpose and, in particular, it might be even a better choice than $\rho_{\max}$ if the dependence studied is non-functional. The MIC suits for situations where there is very little noise in data and, while it has some sort of equitability property, it is not so clear in these R-simulations than in the existing literature.

3.5 Number of observations

Finally, let us yet briefly consider how the number of observations affects the values of measures of dependence. Here, we focus on the case of the cubic dependence of the model (3.3), where the value of the parameter of σ is fixed to 0.1. Figure 12 contains one scatter plot of $n = 300$ observations generated from this model and Table 3 the mean values of seven measures of dependence in 1000 simulations with varying number of observations.

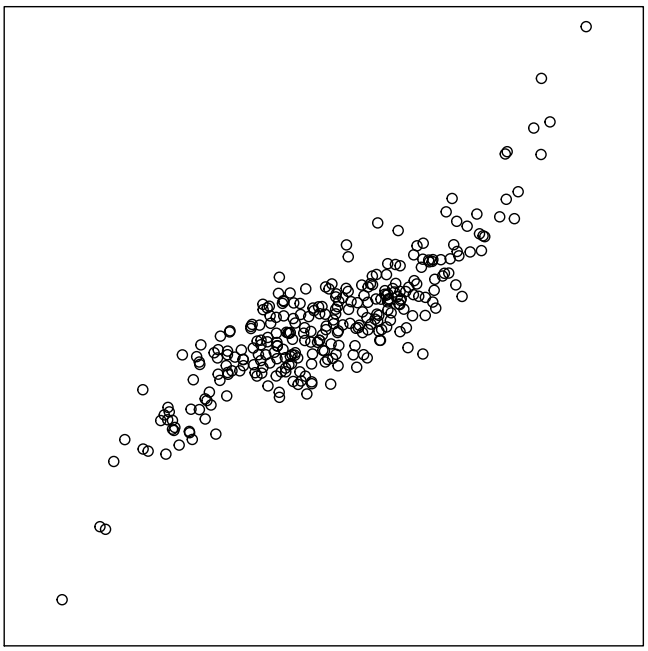


Figure 12: Scatter plot of one simulation from the cubic model (3.3), when $\sigma = 0.1$ and $n = 300$.

Interestingly, we notice that the mean values of the first four coefficients r, r_s, τ and $\rho_{\max}$ are at highest when $n = 30$ and only decrease slightly when the number of observations grows from 30 to 3000. Similarly, the MIC is at highest, too, when $n = 30$ but there are more variation in its mean values. The mean of the mutual information is 0 for $n = 5$ and grows monotonically with respect to n, and so does the informal coefficient of correlation r_1. Furthermore, the order between different coefficients stays mostly the same for each value of n.

To explain these observations, recall firstly the computational methods used. For instance, since the MIC is computed in such grid whose dimensions depend on the value of n with the function $B(n)$, it is quite expected that changing n affects this coefficient. Since we know from the earlier simulations that the value of the MIC drops quickly when the amount of noise increases, this could potentially explain the change in the connection between the MIC and n. Namely, if there are less observations, the noise is not so clear and can be understood as such variation that is directly explained with the model.

Furthermore, our method for computing the mutual information does not work properly if there are just 5 observations because the R-function *discretize* returns the data in just one bin. To change this, we should choose some larger number for the input parameter *nbins*, see [37, p. 4]. The fact that the values of the mutual information grow when with respect to n suggest that they also increase with respect to the number of bins used in discretization. Thus, it might be possible that the values of both the mutual information and the coefficient r_1 are too low in the simulations studied earlier. However, instead of studying the mutual information further with R, we next continue to such methods that should return even more exact values for this quantity.

n	r	r_s	τ	$\rho_{\max}$	MI	r_1	MIC
5	0.793	0.718	0.630	0.809	0	0	0.702
10	0.822	0.748	0.618	0.896	0.229	0.514	0.592
30	0.847	0.792	0.628	0.924	0.359	0.701	0.621
100	0.846	0.801	0.625	0.924	0.419	0.749	0.602
300	0.846	0.804	0.625	0.919	0.509	0.798	0.565
500	0.846	0.806	0.626	0.917	0.535	0.810	0.539
700	0.846	0.806	0.626	0.916	0.558	0.819	0.528
1000	0.846	0.807	0.626	0.916	0.577	0.827	0.514
3000	0.846	0.807	0.626	0.916	0.635	0.848	0.477

Table 3: The mean values of Pearson's, Spearman's and Kendall's correlation coefficients (r, r_s, τ), the maximal correlation coefficient $\rho_{\max}$, the mutual information (MI), the informal coefficient of correlation r_1 and the MIC in 1000 simulations generated from the cubic model (3.3), when $\sigma = 0.1$ and n varies.

4 Neural estimation

In this section, we first summarize the theory and history of neural networks and also explain a few problems related to their use. Then we introduce the MINE algorithm, which can be used to find the mutual information from a large data set with the aforementioned neural networks. At the end, we try out this algorithm and run a few simulations to study its behavior like we did for the different measures of dependence with R.

4.1 Theory

A *neural network* is a highly parametrized model that learns through examples and updates its own parameters accordingly. The idea behind it is vaguely inspired by the human brain: The structure of a neural network can be seen as an artificial presentation of the real, biological neural circuits found in our brains. The concept of neural networks is a very current topic in the study of artificial intelligence, data science and predictive modelling. [16, p. 351]

The history of neural networks began already in the 1940s. Namely, the first model of neural networks was introduced in 1943 by the neurophysiologist W. McCulloch and the mathematician W. Pitts, who used relatively simple logic functions following fixed threshold rules to illustrate how the real neutrons work [36, p. 4]. While some other scientists expressed their interest in this new area of research, there was very little significant progress during the following decades, probably due the computational limitations of the time.

However, several factors contributed to the new interest in the study of neural networks in the 1980s [36, p. 5]. New algorithms, more effective computers and annual conferences organized both by the statistical and computer science communities all promoted the further development of this area of study [16, p. 352]. In spite of this enthusiasm, the usage of neural networks did not become wide-spread during that time. Namely, because even simple applications require neural networks consisting out of an overwhelming number of memory units called *neurons* [56, p. 4], these applications could not be built yet and also the scientific interest on the field quieted down during the mid-1990s [16, p. 352].

Nonetheless, the study of neural networks increased again after 2010s [16, p. 352]. The improved computation resources finally enabled the commercial use of the artificial intelligence applications, such as image classification, text interpretation and speech recognition. Since this awoke not only the interest of the scientific communities but also increased the public attention from the media on the field, there is more funding for the interested scientists. Finding enough data to train neural network is less of an issue in the era of digitalization and there are more work opportunities than ever before. Furthermore, the availability of this field has also increased: Instead of needing a license to install expensive programs and being required to study different programming languages, one can experiment with the readily-built neural network programs on the open-source online platforms.

Let us now explain more carefully how a neural network actually works. Its main principle is the process of *supervised learning* [16, pp. 351-352]. Suppose that we

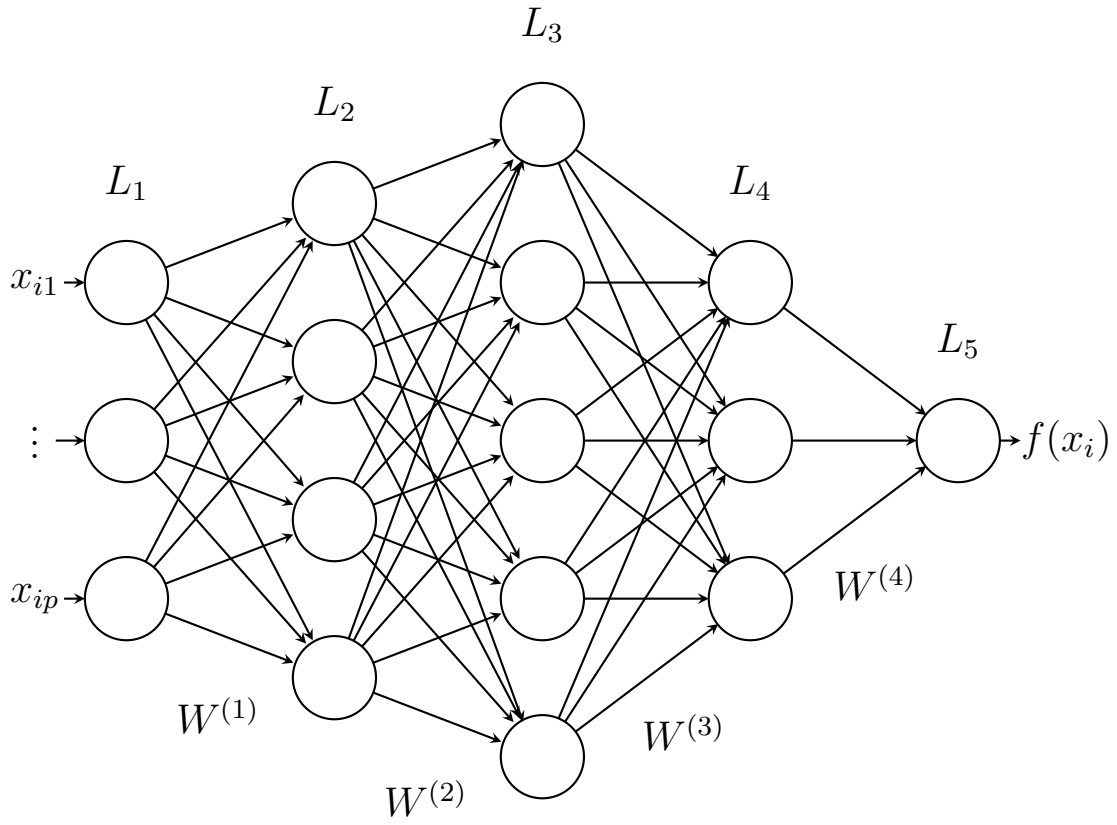


Figure 13: A deep neural network with input layer L_1, three hidden layers L_2, L_3, L_4, and output layer L_5 that, by using the weights W, returns the output $f(x_i)$ for a p-dimensional input vector x_i.

have collected a data set of n observations from the values of the random variables X and Y. In order to oversee how the network works, we need to know how well it performs on this *training set* (x_i, y_i), $i = 1, ..., n$. Here, assume that the components x_i and y_i of the training data are either real values or vectors of real numbers.

Every neural network contains a certain number of neurons, which are connected to each other so that they can transmit and interpret signals, see Figure 13. These neurons are divided into three types of layers: First, there is an input layer L_1, then some number of hidden layers $L_2, ..., L_{N-1}$ and, finally, an output layer L_N [16, p. 352]. If the neural network is composed out of more than one hidden layer or, equivalently, $N > 3$ here, the network is called a *deep neural network* [51, p. 514]. The neurons on each layer L_k, $k = 2, ..., N$, are connected to those of the previous layer via *weights* $\{w_{\ell j}^{(k)}\}$, where ℓj refers to the ℓth neuron and the jth variable [16, p. 352]. The model also contains the intercept term $\{w_{\ell 0}^{(k)}\}$ called *bias* [16, p. 352] and therefore resembles a non-linear statistical model.

Denote now the collection of all the weights in the neural network by W and let x be some input vector given to the network. The ℓth neuron of the $(k-1)$th layer returns the output $z_\ell^{(k)}$ for the next layer $k = 2, ..., N$ so that [16, (18.3) & (18.4),

p. 355]

$$z_\ell^{(k)} = g^{(k)}\left(w_{\ell 0}^{(k-1)} + \sum_{j=1}^{p_{k-1}} w_{\ell j}^{(k-1)} z_j^{(k-1)}\right),$$

where $g^{(k)}$ is some layer-specific transformation, $z_j^{(1)} = x_j$ and $p_1 = p$. The function $g^{(k)}$ can be the identity function in some very simple networks, but it can also be some more complicated non-linear transformation. The final output of the whole network is the output of the formula above for $k = N + 1$.

Clearly, the network described above could be presented with some complex function $f(x; W)$. To determine how well the network performs on the training data, we use a *loss function* $L(y, f(x; W))$ that tells the difference between the real observations y and the output values $f(x; W)$ of the neural network. Our aim is to find such values for the weights that the value of the loss function is at minimum or, equivalently, solve [16, (18.8), p. 356]

$$\underset{W}{\text{minimize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i; W)) + \lambda J(W) \right\}, \tag{4.1}$$

where λ is a *tuning parameter* and $J(W)$ is some nonnegative regularization term of the weights W. [16, pp. 353-356]

If the loss function $L(y_i, f(x_i; W))$ and the regularization term $J(W)$ are suitably chosen, the weights W in (4.1) can be found with differentiation. Namely, if we compute the vector of the first partial derivatives of $L(y_i, f(x_i; W))$ with respect of all the weights in W, this *gradient* tells us the direction where the sum expression in (4.1) decreases the most rapidly [18, Cor. 2, p. 78]. Since $f(x_i; W)$ is defined as a series of compositions on the layers L_k, we can consider the gradients of the weights $W^{(k)}$ of every layer separately and use the following *gradient descent method* for each $k = 1, ..., N - 1$ [16, p. 358]: Choose some initial weights and update them as in [16, (18.16), p. 358]

$$W^{(k)} \leftarrow W^{(k)} - \alpha(\nabla W^{(k)} + \lambda \frac{\partial J(W)}{\partial W^{(k)}})$$

where [16, (18.17), p. 358]

$$\nabla W^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial L(y_i, f(x_i; W))}{\partial W^{(k)}}$$

and $\alpha \in (0, 1]$ is the *learning rate*, until the weights $W^{(k)}$ giving the minimum in (4.1) are found.

However, using the method introduced above can be computationally laborious if the number n of observations in the training data (x_i, y_i) is large. In this case, it is more effective to choose a *batch size* smaller than n and use the training data for the gradient steps separately in batches of this size [16, pp. 358-359]. When the number of observations processed in these batches would be equivalent to the original data size n if combined together, one *epoch* is completed [5, p. 6]. Typically,

solving (4.1) can take hundreds or even thousands of epochs and the convergence of the expression to be minimized is visually studied by plotting its values against the number of epochs, see for instance [53, Fig. 1, p. 8]. The batch size, number of epochs, learning rate and other parameters can be specified by the code when building a neural network [22].

When studying neural networks, one must also note that there are several issues with them. Firstly, because these models rely on supervised learning, they require large amounts of suitably labeled data [24, p. 252]. If there is not enough training data, overfitting might occur and it impairs the predictive accuracy [38, p. 104]. Instead of checking what happens on the hidden layers of the neural network, the systems are often treated as a "black box", which can also result in illogical structures in the weights [38, p. 104]. Furthermore, there is no guarantee that the gradient descent method finds the global minimum for (4.1), because if there is some other local minimum that this method finds first, the network parameters will not be updated anymore and cannot therefore locate the correct minimum.

Another issue is that the neural network can only be as good as the training data. For instance, suppose a neural network is built to find suitable candidates for a job by using CVs of the former applicants and information about whether they got the job or not. If the person who originally selected the new employees was prejudiced against certain minorities, then the neural network will become similarly biased and use such criteria for its decisions that is irrelevant to a person's capability to do well in the job.

Furthermore, if the topic studied is very complex, such as the human language, neural networks are unlikely to recognize all the subtleties without guidance from a human [24, p. 251]. Unlike artificial networks, humans have previous experiences that they can use to help process new information and, if they find a similar logic as in some subject that they are already familiar with, learning is faster. The language and many other interesting phenomena also change continuously according to the surrounding world and it is impossible to give the neural network full data about everything that is happening around us.

To conclude, studying neural networks is both theoretically interesting and useful, but there are certain limitations that need to be taken into account. Even though these kinds of systems become more and more common all the time, we cannot assume that they will give us absolutely correct answers. While neural networks can help to find possible solutions, it is important to interpret the results carefully and check if they are logical before implementing them into practice.

4.2 MINE algorithm

In order to compute the mutual information with the R-function used earlier, one needs to first discretize the data with a binning algorithm and then find the mutual information of this discretized data. This method clearly does not give exact results and, as we noted earlier, the values of both the mutual information and the information coefficient of correlation might therefore have been too low in our R simulations. However, to solve this issue, we can use the neutral networks presented in the preceding subsection to find the ground truth mutual information.

M. I. Belghazi et al. first introduced the *mutual information neural estimator (MINE) algorithm* in their article [2] published in 2018. This algorithm is used to compute the mutual information between two high dimensional continuous random variables by applying neural networks to find the value of one essentially sharp lower bound [2, p. 1]. In fact, the MINE algorithm uses the gradient descent method presented earlier and could be described with a similar graph as in Figure 13.

The MINE algorithm is not the only possible algorithm for finding the value of the mutual information, but it is in our focus because its theoretical properties are well-justified. Namely, it is strongly consistent [2, Thm 2, p. 4] and, if there are enough observations, any desired accuracy can be achieved [2, Thm 3, p. 4]. Furthermore, the MINE algorithm has already been studied further by several scientists, see for instance [11, 13, 30, 45], despite the fact that it was introduced only three years ago.

To understand the structure behind the MINE algorithm, recall the definition of the Kuhlback-Leibler divergence (2.14), its dual presentation (2.15) and its connection to mutual information (2.16). It follows from these results that the mutual information between the random variables X and Y is

$$I(X;Y) = D_{\mathrm{KL}}(P_{XY} \| P_X \otimes P_Y) = \sup_{T:\Omega\to\mathbb{R}} \left(\mathrm{E}_{P_{XY}}(T) - \log(\mathrm{E}_{P_X\otimes P_Y}(e^T))\right), \tag{4.2}$$

where $T : \Omega \to \mathbb{R}$ denotes all such functions that give a finite value for the expression above, $\mathrm{E}_{P_{XY}}$ is the expected value taken with respect to the joint distribution of X and Y, and $\mathrm{E}_{P_X\otimes P_Y}$ is the expected value taken over the product of the marginal distribution of these variables. Define then the *neural information measure* [2, (10), p. 3]

$$I_\Theta(X;Y) = \sup_{\theta\in\Theta} \left(\mathrm{E}_{P_{XY}}(T_\theta) - \log(\mathrm{E}_{P_X\otimes P_Y}(e^{T_\theta}))\right),$$

where the supremum is taken all functions $T_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ parametrized by a deep neural network, $\mathcal{X}$ and $\mathcal{Y}$ are the value sets of X and Y, and Θ is the set of the parameters θ of the neural network considered. Thus, by the equality (4.2), this is a lower bound for the mutual information $I(X;Y)$ [2, (9), p. 3];

$$I(X;Y) \geq I_\Theta(X;Y).$$

The MINE algorithm estimates this lower bound: Given n observations of the continuous random variables X and Y, it returns the *mutual information neural estimator (MINE)* [2, (11), p. 3]

$$\widehat{I(X;Y)}_n = \sup_{\theta\in\Theta} \left(\mathrm{E}_{P_{XY}^{(n)}}(T_\theta) - \log(\mathrm{E}_{P_X^{(n)}\otimes P_Y^{(n)}}(e^{T_\theta}))\right) \tag{4.3}$$

as an output. Here, $P_X^{(n)}$ denotes the empirical distribution of the variable X, which can be estimated from the data. In order to estimate the distributions needed for (4.3), the number n of observations naturally needs to be quite large but it is noteworthy that the MINE algorithm is built for this exact purpose of finding the mutual information from large data sets.

The attached Algorithm 1 contains the pseudocode from [2, Alg. 1, p. 3], which tells us in more detail how the MINE algorithm actually finds the value of MINE.

Algorithm 1 MINE

$\theta \leftarrow$ initialize network parameters
Repeat
  Draw a batch of b samples from the joint distribution:
  $(x^{(1)}, y^{(1)}), ..., (x^{(b)}, y^{(b)}) \sim P_{XY}$
  Draw n samples from the marginal distribution of Y: $u^{(1)}, ..., u^{(n)} \sim P_Y$
  Evaluate the lower bound:
  $V(\theta) \leftarrow \frac{1}{b}\sum_{i=1}^{b} T_\theta(x^{(i)}, y^{(i)}) - \log(\frac{1}{b}\sum_{i=1}^{b} e^{T_\theta(x^{(i)}, u^{(i)})})$
  Evaluate bias corrected gradients: $\hat{G}(\theta) \leftarrow \nabla_\theta V(\theta)$
  Update the network parameters: $\theta \leftarrow \theta + \hat{G}(\theta)$
Until convergence of the lower bound $V(\theta)$

As mentioned before, the idea of this algorithm is based on the gradient descent method, which is used to find the supremum in the expression (4.3) of MINE. Note also that the MINE algorithm uses batches of size b instead of the whole data in order to compute this quantity effectively. While the gradient computed from these batches would be biased when compared to the gradient obtained by using the original data of size n, the MINE algorithm takes this into account by using bias corrected gradients [2, p. 3].

There are a few possibilities for the implementation of the MINE algorithm. In this work, we use the library PyTorch, which is a free open-source software that allows the user build neural networks with Python or C++. While neural networks could be built with the R-package *neuralnet* [21], there are several ready-written PyTorch codes for the MINE algorithm that can be downloaded from the software development and hosting site GitHub. For instance, in 2018, M. Yamada released a code [54] for the PyTorch implementation of the MINE algorithm based on the original article [2].

While the MINE algorithm is easy to understand, there are some issues with the estimator MINE, which is given as an output of this algorithm. Namely, even though MINE has some good theoretical qualities such as high accuracy, it is based on the assumption that there is an unlimited supply of data to avoid overfitting [30, p. 1]. If this not the case, some other estimator should be used instead.

In their work [45] from 2019, J. Song and S. Ermon proposed a set of three *self-consistency* tests for the estimators $\hat{I}(X;Y)$ of the mutual information between variables X and Y. According to them, the value of $\hat{I}(X;Y)$ should be 0 if X and Y are independent, $\hat{I}(X;Y) \approx \hat{I}([X, f(X)]; [Y, g(Y)])$ for all functions f, g such that $\hat{I}(X;Y) \geq \hat{I}(f(X); g(Y))$, and $\hat{I}([X_1, X_2]; [Y_1, Y_2]) \approx 2\hat{I}(X;Y)$ if X_1, X_2 are independent random variables with the same distribution as X and Y_1, Y_2 defined similarly for Y [45, pp. 5-6]. Here, $[,]$ denotes concatenation. However, neither the estimator MINE defined (4.3) nor any other known estimator has all these three properties [45, p. 6]. Furthermore, according to Song and Ermon, the values of the MINE can also sometimes exhibit variance that grows exponentially with respect to the real value of the mutual information [45, p. 5].

Thus, the MINE algorithm can be used to compute the value of the mutual information with neural networks. This method should be effective and it can be

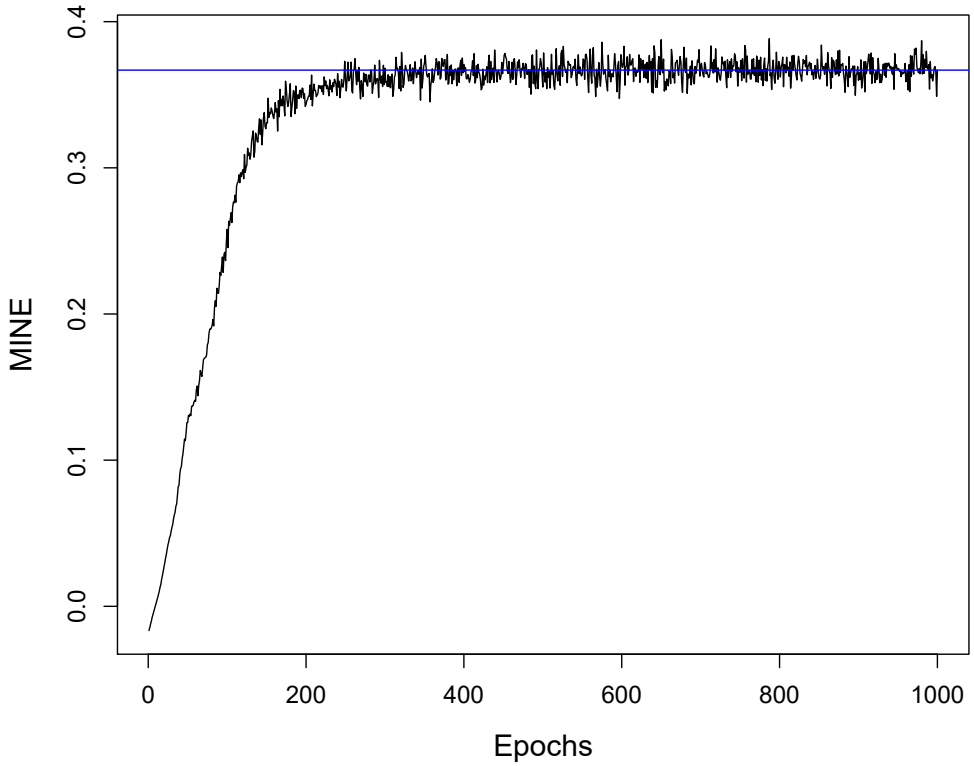


Figure 14: The outputs of the MINE algorithm on 1000 epochs for data simulated from the linear model (3.2) with $n = 30000$ and $\sigma = 0.3$, and the mean value of the last hundred outputs (as a blue line).

implemented from already existing codes by using open-source software. Still, even though the estimator behind the MINE algorithm is strongly consistent and accurate if there are enough data, it is only intended for large data sets.

4.3 Results of simulations

Next, we investigate mutual information by computing its values through neural estimation in a few different simulations. We use a slightly modified version of the PyTorch implementation [54] of the MINE algorithm and run this code in the browser-based platform Colaboratory developed by Google. Our simulated data sets are created according to the models (3.1)-(3.5), and we vary the number n of observations and the value of the noise parameter σ to see their effect on the MINE.

Firstly, we must find out how many epochs of data are needed to process until the value of the MINE given as an output by the MINE algorithm converges. For this, we consider data of $n = 30000$ observations generated from the linear model (3.2) with $\sigma = 0.3$. Figure 14 depicts the output of the MINE algorithm for 1000 epochs together with the mean value of the last hundred outputs.

As can be seen from Figure 14, 1000 epochs is clearly enough for the convergence. In fact, the convergence occurs already by the 400th or so epochs, and after that the outputs vary around the mean of the final hundred values. Consequently, if we suppose that the converged value of the estimator MINE is the same as this mean

n	MINE
100	0.3446
1000	0.3571
3000	0.3683
10000	0.3666
30000	0.3668
100000	0.3704

Table 4: The value of the MINE computed as a mean value of the last hundred outputs out of 1000 epochs, when the data is generated from the linear model (3.2) with $\sigma = 0.3$ and the number n of observations varies.

value, the MINE algorithm tells us that the value of the mutual information for this data set is 0.367.

With a few tests run on the MINE algorithm, it can be seen that the number n of observations does not affect the number of epochs needed to the convergence of MINE, at least in the case where the data is generated from the linear model (3.2) with $\sigma = 0.3$. Namely, the convergence happens around 400 or so epochs, regardless of whether n is 100 or 100000. However, there is much more variation in the outputs of the MINE algorithm for small values of n: As we can see from Figure 14, the MINE is quite close to the marked mean line with all of the epochs after the convergence when $n = 30000$, but the corresponding outputs can vary from 0 to 0.8 if we fix $n = 100$ instead.

As we can also see from Table 4, the value of the MINE computed as a mean value of the outputs on the epochs from 901 to 1000 is not very much affected by the changes in the data size n. It can be observed that the values of the MINE increase with respect to n, but this growth is very slight. Raising the number n of observations without changing the batch size means that there are more iterations during which the network parameters are updated, so it is quite natural that the outputs on the final epochs are on average higher because they are closer to the exact point of the convergence.

We can also calculate that if the value of the mutual information is 0.367, then the information coefficient of correlation r_1 is 0.721. From Figure 8, we notice that this value of r_1 is nearly the same as its mean value computed with the R-functions from 1000 simulations with the similar data. Interestingly, this value 0.721 is in fact little less than the coefficient r_1 should be according to Figure 8, which might suggest that more epochs are needed before computing the mean of the MINE.

Next, let us consider the model (3.1) that has no dependence between the variables X and Y. As we recall from the theoretical properties of mutual information, its value should be 0 for any data generated from this model. If we simulate data of $n = 30000$ observations and fix the number of epochs to 1000, the mean value of the MINE computed from the last hundred outputs is $-4.869 \cdot 10^{-6}$. Note that the value of this estimator can be less than 0 because, technically, it is based on a lower bound of mutual information. Compared to the mean value of the mutual information presented in Table 1 for the same model, we can conclude that this result is closer to 0 and therefore more realistic. In other words, the MINE passes

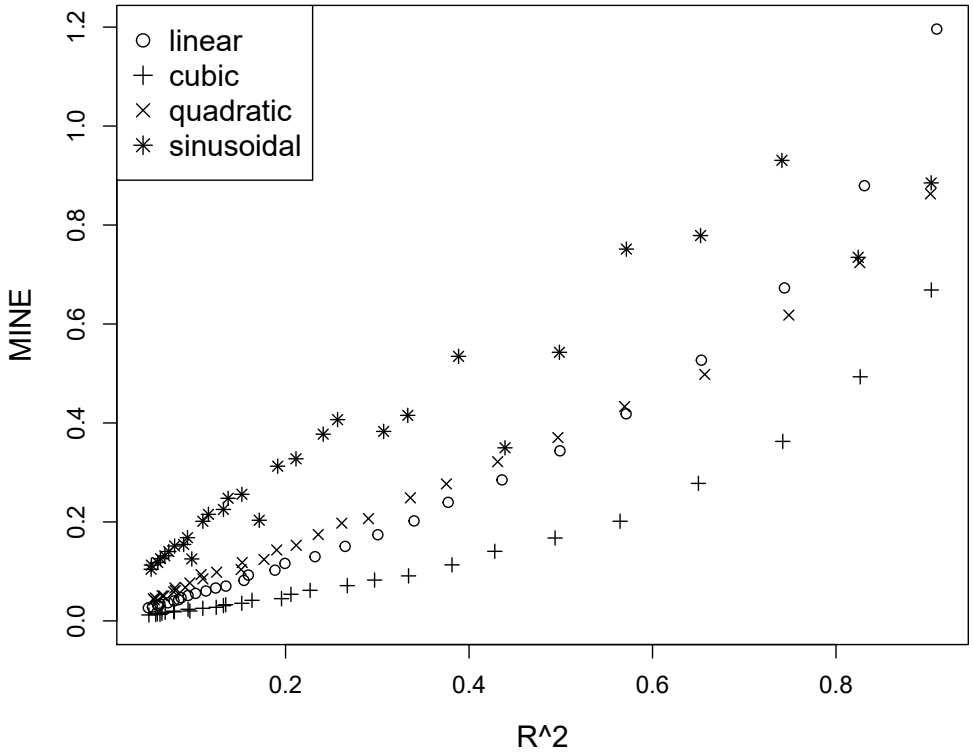


Figure 15: The values of the MINE and R^2 in 30 simulations of the linear, cubic, quadratic and sinusoidal models (3.2)-(3.5) with $n = 30000$ observations.

the first self-consistency test proposed in 2019 by Song and Ermon [45, p. 5].

Let us yet inspect how the amount of noise affects the linear model (3.2), the cubic model (3.3), the quadratic model (3.4) and the sinusoidal model (3.5). For each model, we choose 30 different values of the parameter σ, generate data with $n = 30000$ observations, compute first the coefficient of determination R^2 and then estimate the value of the MINE by taking the mean value of the last hundred outputs of the MINE algorithm run with 3000 epochs. This number of epochs is chosen here because, according to a few tests, it is enough for the convergence of the MINE without slowing the algorithm down too much. The results are depicted in Figure 15.

From Figure 15, we can see that the values of the MINE mostly increase with respect to the value of R^2 or, equivalently, decrease with respect to the amount of noise in each of the four models. The few observations of the sinusoidal model not following this pattern can probably be explained with discrepancies in the convergence of the parameters in the MINE algorithm for this model. By comparing the values of the MINE between the models, we also notice that the values of the MINE computed from the linear model are the greatest if there is very little noise but, as the amount of noise grows and R^2 decreases, the MINE values computed from the sinusoidal and quadratic models exceed those from the linear model.

Interestingly, we also see from Figure 15 that the mutual information estimated with the MINE does not have a similar equitability property as proposed for the MIC in [39, p. 1518]. Namely, for all values of R^2, the MINE values computed from different models can be easily distinguished. However, it must be noted here that

Figure 15 cannot be directly compared to, for instance, Figure 11: We need to take into account that Figure 11 contains results from 1000 simulations with $n = 1000$ observations for each model, while Figure 15 has considerably less simulations but 30 times more observations in each simulation. Furthermore, it is also difficult to create simulations with similar values of n to make these figures comparable. This is because computing the MIC with R is very slow if the value of n grows up to several thousands, but there is too much variation in the outputs of the MINE algorithm with $n = 1000$ or less.

However, while Song and Ermon mentioned the possibility of such variance in the values of the MINE that would increase exponentially with respect to the ground truth mutual information [45, p. 5], I did not notice this phenomenon in any of these simulations. Namely, if the other parameters related to the MINE algorithm had constant values, then decreasing the value of σ did not increase the variation in the outputs of the algorithm, even though this change clearly increases the true value of the mutual information. Instead, the number n of observations was the only factor that had an impact on the variance in the MINE.

To conclude, the MINE algorithm works well for estimating the value of mutual information from simulated data. The estimator MINE recognizes dependence from each data sets data based on different models and its values decrease in an expected way when the amount of noise grows. If there are at least 30000 observations, the outputs of the MINE algorithm converge very clearly, but there is much more variation in these outputs if there are not enough observations.

5 Real data experiments

In this section, we study the different measures of dependence with real data sets about the weather, youth behavior and air pollution. Because some statistical concepts such as the MIC have been developed by using mostly just simulated data, it is important to verify that these quantities work in an expected way also for real data. To do this, we download each data set into RStudio and compute the values for several measures of dependence by using the computational methods introduced in Subsection 3.1.

5.1 Weather in Nuorgam

Our first real data experiment will be done by using recent weather observations. This is because data about weather is continuously collected and shared openly to the public, there are several interesting variables related to it, and the relationships between these variables can be understood without deep expertise in the field. Our data is from Finnish Meteorological Institute, Ilmatieteenlaitos [27].

Out of all possible observation stations in Finland, I chose one in Nuorgam, Utsjoki. Since this station is located in the northernmost point of the country, there should be clear seasonal variation in the data and we can also inspect the amount of snow. However, note that while Ilmatieteenlaitos collects data also about such variables as cloud amount, precipitation amount and horizontal visibility, these are not observed in Nuorgam and will not be therefore studied here, either.

Instead, we consider the following eight variables: Mean sea level air pressure (hPa), relative humidity (%), snow depth (cm), air temperature (°C), dew point temperature (°C), wind direction (an angle in degrees), gust speed (m/s) and wind speed (m/s). Here, the relative humidity means the absolute amount of water vapor in the air relative to its maximum amount in the same temperature, and the dew point temperature is the temperature cool enough that the current amount of water vapor would begin to condense to liquid water. The wind direction is the magnitude of the clockwise angle from the North to the direction from which wind blows, while the gust speed is the speed of sudden wind gusts that are considerably stronger than the average wind but only last a very brief time. All of these quantities are expressed in either integers or decimal numbers with one digit.

Before studying the data itself, let us make a few hypotheses about which variables might be connected. Clearly, air temperature affects the relative humidity and the dew point temperature. Namely, by definition, the dew point temperature is always at most the current temperature. Also, while temperature might not have a direct impact on the snow depth, we know that the amount of snow is typically at highest during the coldest months of the year.

Importantly, the air pressure has a known and very well-studied effect to the weather. Namely, if the pressure is low, the weather is often windy, rainy and possibly even stormy. In turn, during an anticyclone caused by high atmospheric pressure, the air is typically drier and the sky is less cloudy, and the weather can be very warm if it is summer. In other words, the air pressure affects the humidity, the both temperature measurements as well as the both wind speeds.

Furthermore, the wind and gust speeds are also clearly connected because the definition of the gust speed is based on the amount of the average wind speed. Potentially, the wind direction and speed might also have some sort of correlation, especially if the value of the wind direction is fixed automatically to 0 degrees if no wind is observed. It is also known that wind blows stronger over sea than land, which might mean that the wind direction could have an impact on its speed. On top of that, the wind direction might be connected to the air temperature because wind from the North or the sea is often colder. Note that the closest sea area to Nuorgam is Varanger Fjord at 70-80 degrees from the North.

One thing that must be noted here is also the indirect relationships between the variables caused by some third variable. If snow depth and wind direction have both some relationship with air temperature, there is some indirect connection between them. However, these types of relationships are not often strong and might be difficult to recognize if there is already a lot of variation in the data due to other factors.

The data studied here contains the hourly weather observations of the year-long time period from June 1st, 2020, to May 31st, 2021, downloaded from `https://en.ilmatieteenlaitos.fi/download-observations`. As mentioned before, we only focus on the eight variables listed above and, when all the rows with missing observations are removed, the data contains 8751 different observations. While these variables could also be studied against time, it would require taking into account the seasonality, which is difficult with just data from just one year. Thus, we do not consider time at all.

By drawing histograms from the values of our eight different weather-related variables, we notice that the distributions of several variables are negatively skewed. For instance, because the relative humidity is measured as a percentage and it can have values considerably smaller than its mode at 90% but not much higher, it is clearly skewed to the right. Furthermore, the temperature in Utsjoki varies from -40°C to 25°C but, since its mode is around 3°C, the left tail of the distribution is noticeably longer. The distribution of the dew point temperature is a little less skewed than that of regular temperature but its left tail is still longer. Similarly, the distribution of the air pressure is negatively skewed, too, even though there is no obvious explanation for this.

In turn, the snow depth and both of the wind speed measurements are positively skewed because these measurements can only have non-negative values but are often 0 or close to it. Still, while the modes of snow depth, gust speed and wind speed are very small, these variables can have values as high as 56cm, 17.5m/s and 10.9m/s, respectively. Interestingly, the wind direction has multiple different peaks out of which one is around 0 degrees, another at 75 degrees and one around 250 degrees, as can be seen in the histogram of Figure 16. Recall here that 0 degrees means north wind whereas 75 degrees indicates that the wind is from the sea.

However, it would seem that the wind measurements in the data do not fully correspond with the actual observations. Namely, we can check that, out of all the 8751 observations about the wind direction, 749 values are 0 degrees so at least some of these observations are very likely made by default whenever the wind speed is 0.0 m/s instead of there actually being so much north wind. Typically, the wind

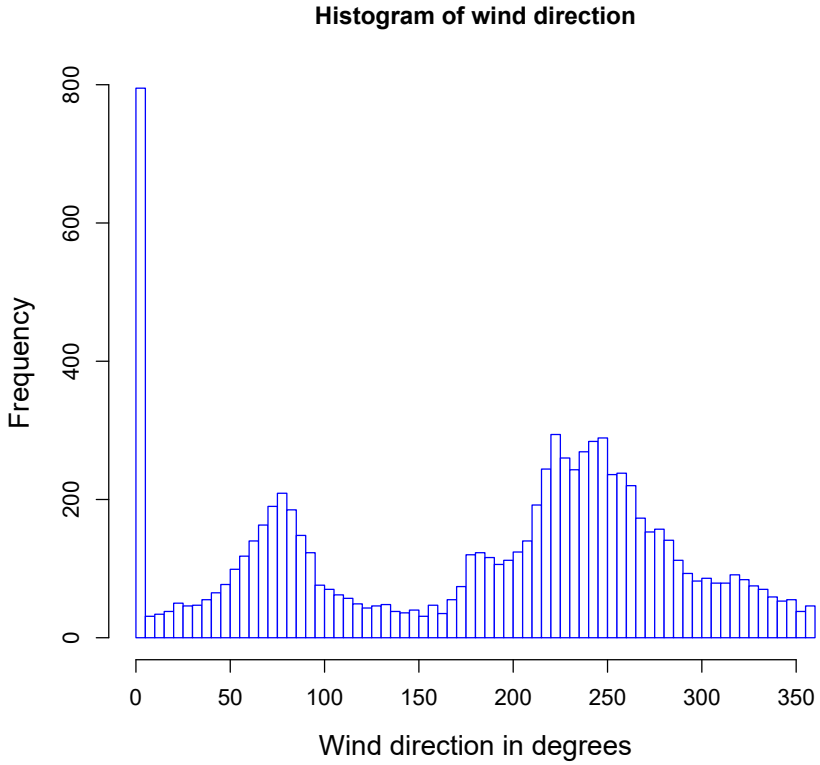


Figure 16: The histogram of the wind direction measured in degrees from the weather data in Nuorgam.

direction should be input as a missing observation (NA in R) whenever there is no wind but let us not change this to see how this issue affects our results. In all likelihood, the data might suggest that there is positive correlation between the wind direction and speed because their smallest values occur nearly always together with the exception of the case where there is noticeably north wind.

Let us now focus on the correlation coefficients. Recall the two assumptions related to Pearson's correlation: The values of the variables should be approximately normally distributed and their dependence needs to be linear so that it can be properly recognized. Because none of our variables is normally distributed, we should consider either Spearman's or Kendall's correlation coefficient instead. Since the values of Spearman's correlation coefficient are very similar to Kendall's coefficient for this data, but the former coefficient is more commonly used, we use it to study the correlation here. The values of Spearman's correlation coefficient and other measures of dependence considered here are collected in Table 5.

Firstly, we can notice that most of the values of Spearman's correlation coefficient between these eight variables are negative but not by much. For instance, by considering the values of Spearman's coefficient between the air pressure and the other variables, we notice that the value -0.21 against the dew point temperature is the one furthest away from 0, so the air pressure is very weakly correlated with the other weather measurements. Similarly, there is not much correlation between the relative humidity and the other variables but, interestingly, the values of Spearman's correlation coefficient between the relative humidity and both the wind speed

Weather variables	r_s	$\rho_{\max}$	MI	r_1	MIC
Pressure; humidity	-0.176	0.189	0.053	0.318	0.067
Pressure; snow depth	0.024	0.276	0.142	0.498	0.124
Pressure; temperature	-0.143	0.380	0.160	0.523	0.110
Pressure; dew point	-0.206	0.366	0.166	0.531	0.108
Pressure; wind dir.	-0.064	0.179	0.061	0.339	0.051
Pressure; gust speed	-0.183	0.188	0.049	0.306	0.062
Pressure; wind speed	-0.168	0.175	0.046	0.297	0.059
Humidity; snow depth	-0.080	0.316	0.139	0.493	0.096
Humidity; temperature	-0.165	0.475	0.215	0.591	0.133
Humidity; dew point	0.094	0.280	0.156	0.517	0.093
Humidity; wind dir.	-0.173	0.232	0.083	0.392	0.072
Humidity; gust speed	-0.441	0.445	0.149	0.507	0.146
Humidity; wind speed	-0.427	0.428	0.133	0.483	0.128
Snow depth; temperature	-0.788	0.849	0.620	0.843	0.609
Snow depth; dew point	-0.808	0.864	0.660	0.856	0.656
Snow depth; wind dir.	0.007	0.138	0.080	0.385	0.057
Snow depth; gust speed	0.004	0.150	0.040	0.278	0.038
Snow depth; wind speed	0.021	0.180	0.045	0.294	0.041
Temperature; dew point	0.950	0.984	1.456	0.972	0.759
Temperature; wind dir.	0.095	0.246	0.106	0.436	0.072
Temperature; gust speed	0.243	0.403	0.110	0.444	0.110
Temperature; wind speed	0.224	0.394	0.110	0.444	0.106
Dew point; wind dir.	0.036	0.210	0.095	0.417	0.060
Dew point; gust speed	0.144	0.363	0.100	0.426	0.099
Dew point; wind speed	0.126	0.355	0.102	0.430	0.095
Wind dir.; gust speed	0.339	0.956	0.360	0.717	0.422
Wind dir.; wind speed	0.325	0.955	0.364	0.719	0.422
Gust speed; wind speed	0.969	0.967	1.393	0.969	0.776

Table 5: Spearman's correlation coefficient r_s, the maximal correlation coefficient $\rho_{\max}$, the mutual information (MI), the informal coefficient of correlation r_1 and the MIC for different weather variables including mean sea level air pressure, relative humidity, snow depth, air temperature, dew point temperature, wind direction, gust speed and wind speed.

measurements are around -0.43.

Several of the values of Spearman's correlation coefficient are as we expected. For example, there is negative correlation between the snow depth and the two temperature variables: Spearman's coefficient is -0.79 between the snow depth and the air temperature, and -0.81 between the snow depth and the dew point temperature. Similarly, it is not surprising that the value of Spearman's correlation coefficient between these temperature variables is 0.95, which is the second largest value here even when considering only the absolute values for this coefficient. The largest value of Spearman's coefficient for these variable pairs is 0.97 between the gust speed and the wind speed, which is a very expected result, too. It is also worth noting that the

correlation between the wind direction and both of the wind speeds is 0.33, which is quite large when compared to other values here and very likely due to the observation we made earlier about how the values of 0 of the wind direction and speed occur together.

Out of all these variable pairs, Spearman's correlation coefficient suggests that the weakest relationship is between the snow depth and the gust speed, for which the value of this correlation coefficient is $3.94 \cdot 10^{-3}$. This is actually quite understandable: While the air pressure might be connected with both the air temperature and how windy it is and the temperature has a clear relationship with snow depth, the causalities here are not so immediate that they would show in the data. Also Spearman's coefficient is quite close to 0 for the relationship between the snow depth and the wind speed for the same reason.

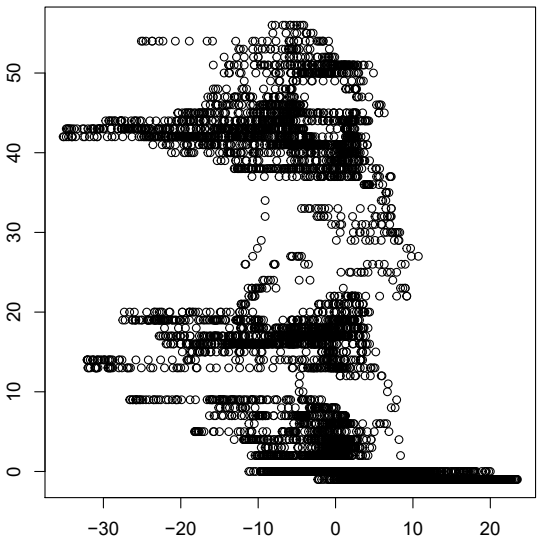
Next, let us consider the values of the maximal correlation coefficient for these variables. These values do not vary very much from those of Spearman's correlation coefficient, except they are all positive and slightly larger. In fact, by denoting Spearman's coefficient by r_s, then most of the values of the maximal correlation are on the interval $[|r_s|, \min\{|r_s| + 0.2, 1\}]$. However, there are a few exceptions: The maximal correlation between the relative humidity and the air temperature is 0.48, even though Spearman's coefficient is only -0.165 for these variables, and the maximal correlation between the wind direction and both of the wind speed measurements is over 0.95, in spite of the fact that the corresponding values of Spearman's coefficient are around 0.33, as mentioned above.

Then we compute the values of the mutual information, which vary from 0.040 to 1.456. The smallest value of the mutual information is between the snow depth and the gust speed, whereas the largest values are between the two temperature measurements and also for the two wind speeds. After transforming the values of the mutual information to those of the information coefficient of correlation r_1, we notice that the values of r_1 are very often clearly larger than the maximal correlation: For instance, while the maximal correlation between the air pressure and the snow depth is 0.28, the corresponding value of r_1 is over 0.5.

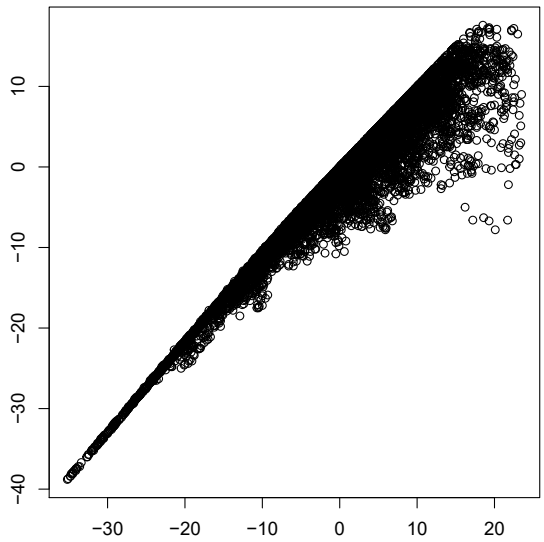
Finally, we consider the values of the MIC. We notice that they are very small for such variables that are clearly connected and, for instance, the MIC is only around 0.75 for both the variable pair consisting of the temperature measurements and the pair of the wind speeds. Similarly, the MIC is 0.61 between the snow depth and the air temperature, which is less than the value of r_1, the maximal correlation and the absolute value of Spearman's correlation coefficient between these two variables. However, according to the values of the MIC, the weakest relationship is between the snow depth and the gust speed, which agrees with the results given by the other measures of dependence.

Some of these observations related to the different values of dependence can be easily explained. For instance, both in our earlier simulations and in the literature [35, 40, 43], it has been noticed that the MIC is very sensitive to the statistical noise. Thus, since the relationships of the real word, related to the weather or not, contain always some noise, it is to be expected that the values of the MIC are less than those of the other quantities considered.

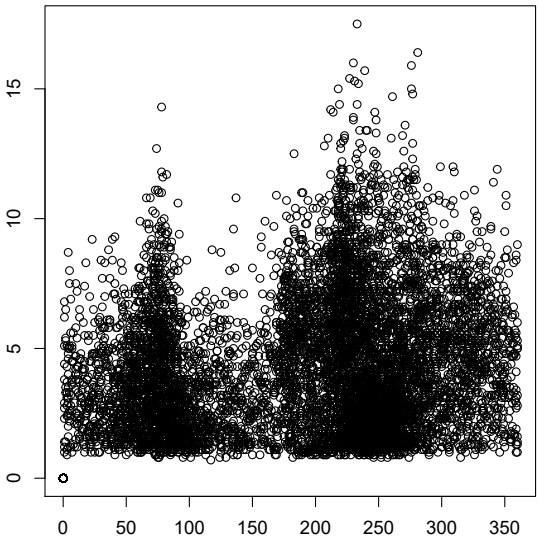
Plotting the different relationships as in Figure 17 also helps us to explain the



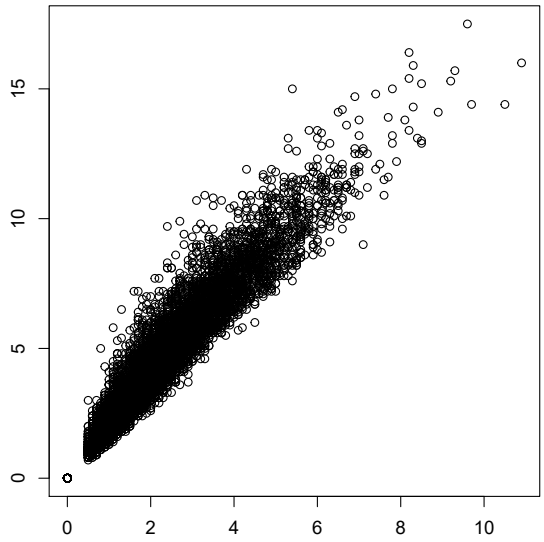
(a) Snow depth against air temperature



(b) Dew point temperature against air temperature



(c) Gust speed against wind direction



(d) Gust speed against wind speed

Figure 17: Scatter plots between a few different variables in the weather data.

differences in the values of these measures of dependence. For instance, we see from Figure 17d that the relationship between the gust speed and the wind speed is linear, which explains why the values of Spearman's correlation coefficient and the maximal correlation are nearly the same for this variable pair. While the connection between the snow depth and the air temperature shown in Figure 17a is not monotonic, it does not follow any other function much better, which is why the value of the maximal correlation is not much higher than the absolute value of Spearman's correlation coefficient.

We also noticed earlier that the maximal correlation between the wind direction and both of the wind speeds is considerably higher than Spearman's coefficient. There is no linear or non-linear relationship clearly visible in Figure 17c but, as noted before, these variables have values of 0 at the same time. If we remove all such observations from our data for which the wind speed is 0 m/s, the maximal correlation is only 0.216 between the wind direction and the gust speed and 0.191 between the wind direction and the wind speed instead of 0.956 and 0.955.

Consequently, this leads to a question about the trustworthiness of the maximal correlation between two variables with very uneven distributions. Namely, the maximal correlation is computed by finding the transformation that maximizes Pearson's correlation coefficient and, even though the variables need to be approximately normally distributed when computing correlation with just Pearson's coefficient, there is no such assumption related to the maximal correlation. The natural solution to fix this issue would be to use the same transformation as usually to maximize the linear correlation but then compute the value of Spearman's correlation coefficient instead to obtain the maximal correlation. This would give the values 0.324 and 0.334 for the maximal correlation between the wind direction and the two wind speeds in the data with all the available original observations, which would seem a more realistic result.

To conclude, our first real data experiment went as expected. All the measures of dependence behaved as they did in our earlier tests with the simulated data sets and there were very few surprises in the relationships found between different weather variables. The only question emerged here is whether the definition of the maximal correlation could be slightly modified so that this quantity would fit better for relationships between such variables that do not follow a normal distribution.

5.2 Youth risk behavior

Our next experiment is about such behavior of teenagers that is considered unhealthy or risky for their health. This type of data is very different from the weather observations and, for instance, consists of mostly ordinal variables instead of numerical ones. Thus, we can potentially find new information about the behavior of our measures of dependence that is not noticeably when studying quite straightforward relationships between weather variables.

The data considered here is collected by Centers for Disease Control and Prevention (CDC), which is a national public health agency in the United States. For several years, CDC has conducted different surveys that monitor health-related behaviors of teenagers that could lead to disability or deaths. The Youth Risk Behavior

Surveillance System (YRBSS) was developed in 1990 to collect data about the habits of middle and high school students, such as the amount of their physical activity, sleep and drug use. [8]

The data we use here is the data set called *yrbss* from the R-package *openintro* [9]. While it is confirmed in the information about this data set that its source is CDC's YRBSS [9, p. 240], it is not specified exactly when this data is collected. The package *openintro* was published in April, 2021, but YRBSS has not yet released the data of this year [8]. Since there are 13583 observations on the data set and YRBSS has obtained around 14000±1000 usable observations after editing every other year from 1991 to 2019 [8], this data could be from a single year. Furthermore, I would guess that the data is from the 90s or the 00s rather than from a more recent year because none of the variables in the data are related to the use of Internet, smart phones or social media but nearly all of the teenagers surveyed here spend at least 3 hours daily watching television on a school night.

Out of all 13 variables of *yrbss*, I only consider the following nine: gender (male or female), age (in years), height (m), using a helmet while riding a bike in the last year (did not ride, never, rarely, sometimes, most of the time, or always), days on which texted while driving in the last month (did not drive, 1-2, 3-5, 6-9, 10-19, 20-29, or 30), the number of days with at least one hour of physical activity during the last week, the amount of TV watched on a typical school night (do not watch, <1, 1, 2, 3, 4, 5+), the number of days with strength training during the last week and the hours of sleep on a typical school night (<5, 6, 7, 8, 9, 10+). I turned this data fully numerical by replacing the gender with binary variables, "did not drive" and "did not drive" with missing observations (NA in R) and the other options for the helmet use variable with numbers so that 0="never", 1="rarely", 2="sometimes", 3="most of the time" and 4="always". I also replaced each interval of days with its mean point and the values <1, 5+, <5 and 10+ with 0.5, 6, 4 and 11, respectively.

Some of these variables are clearly connected. For instance, there is a relation between a person's height and age and, since the data is mostly about underage teenagers, this should be noticeable. Also, there is probably positive correlation between amount of strength training and other physical activity. Furthermore, the amount of TV and sleep on a typical school night are likely negatively correlated because there is not enough time to both watch television for several hours after a typical seven-hour school day with travels, homework and other activity if one sleeps at least 10 hours. However, this data can be used to study also more complicated relationships, such as the one between the amount of TV and wearing a cycling helmet, which might be potentially very interesting.

By plotting the histograms of this edited data, one can see that only the height and sleep variables follow even approximately normal distribution. Since there are also several ordinal variables in the data, it is therefore better to use Spearman's correlation coefficient than Pearson's while measuring the correlation. Like for the weather variables, I computed here Spearman's coefficient, the maximal correlation, the mutual information, the information coefficient of correlation r_1 and the MIC for all the other eight variables except the gender, see Table 6.

From Table 6, we notice that nearly all relationships in this data are very weak. Namely, with only a few exceptions, the absolute values of the most of Spearman's

Variable pair	r_s	$\rho_{\max}$	MI	r_1	MIC
Age; height	0.130	0.146	0.015	0.170	0.016
Age; helmet use	-0.030	0.035	0.002	0.059	0.002
Age; texting	0.296	0.305	0.054	0.320	0.072
Age; phys. activity	-0.064	0.071	0.004	0.087	0.004
Age; TV	-0.029	0.034	0.002	0.061	0.002
Age; strength training	-0.063	0.075	0.005	0.100	0.005
Age; sleep	-0.127	0.126	0.011	0.146	0.011
Height; helmet use	-0.023	0.016	0.003	0.083	0.005
Height; texting	0.112	0.119	0.013	0.163	0.012
Height; phys. activity	0.211	0.211	0.027	0.229	0.028
Height; TV	0.010	0.033	0.005	0.097	0.005
Height; strength training	0.192	0.196	0.023	0.212	0.025
Height; sleep	0.020	0.038	0.003	0.081	0.005
Helmet use; texting	-0.092	0.092	0.006	0.110	0.007
Helmet use; phys. activity	0.027	0.037	0.003	0.078	0.002
Helmet use; TV	-0.094	0.111	0.008	0.128	0.009
Helmet use; strength training	0.016	0.034	0.003	0.079	0.003
Helmet use; sleep	0.092	0.086	0.007	0.116	0.006
Texting; phys. activity	0.046	0.058	0.005	0.103	0.003
Texting; TV	-0.027	0.076	0.007	0.122	0.005
Texting; strength training	0.042	0.070	0.007	0.114	0.004
Texting; sleep	-0.089	0.095	0.010	0.139	0.007
Phys. activity; TV	-0.039	0.122	0.011	0.148	0.011
Phys. activity; strength training	0.621	0.625	0.326	0.692	0.255
Phys. activity; sleep	0.124	0.134	0.014	0.163	0.011
TV; strength training	-0.020	0.101	0.010	0.140	0.007
TV; sleep	0.016	0.140	0.013	0.158	0.010
Strength training; sleep	0.112	0.108	0.010	0.141	0.008

Table 6: Spearman's correlation coefficient r_s, the maximal correlation coefficient $\rho_{\max}$, the mutual information (MI), the informal coefficient of correlation r_1 and the MIC for different variable pairs including age, height, using a helmet while biking, texting while driving, physical activity, watching TV, strength training and sleep.

correlation coefficients are less than 0.20 and neither the maximal correlation nor the coefficient r_1 is much higher than this. The MIC is very close to 0 for all the variable pairs, too. It would also seem that these existing relationships are both functional and monotonic because they cannot be recognized any better with the other coefficients than Spearman's correlation coefficient.

The only noticeable relationships from Table 6 are between physical activity and strength training, the age and texting while driving, the height and physical activity, and the height and strength training. According to the values of Spearman's correlation coefficient, the correlation between all these four variable pairs is positive. However, the physical activity and strength training is the only variable pair for which any of the measures of dependence considered in Table 6 have values over 0.5.

It must be noted here that the noticeable relationships between the height and both physical activity and strength training are not necessarily direct connections but could potentially explained by taking the gender variable into account instead. This hypothesis can be easily tested by computing the values for the different measures of dependence again within two separate subgroups, out of which one consist of only girls while the other one contains all the boys. Since all the coefficients have values very close to 0 for the relationships between the height and physical activity or strength training in these same-sex groups, our guess was correct. In fact, it can be computed from this data that the girls surveyed here have average height of 162cm, are physically active on 3.26 days during a week and train their strength on 3.30 days, whereas the corresponding numbers for boys are 1.76cm, 4.52 and 3.58, respectively.

However, by studying these two same-sex subgroups, we notice that not all relationships can be explained with the gender variable. For instance, there is slightly more correlation between the age and texting while driving in the girl group, but all the coefficients still have very similar values for this relationship between the two groups. There is an easy explanation behind this relationship, though: While the minimum age for getting a learner's permit is 14 or 15 years in the United States, the younger teenagers are unlikely to drive very much and the presence of their driving teacher or guardian might be obligatory. Since the data only shows how many days a teenager texts while driving if they drive at least a little and not how often this occurs in relation with the time spent driving, it is to be expected that older teenagers who drive more, often alone or with their friends, also text while driving more. Furthermore, physical activity and strength training are not affected by gender but their correlation is quite expected.

Interestingly, we can also notice differences between the height growth between the boys and girls in this data set. Namely, all the coefficients have absolute values less than 0.1 between the age and the height inside the girl group but, for the boy group, Spearman's correlation coefficient is 0.219, the maximal correlation 0.251 and the value of r_1 0.259. This can be explained by the fact that girls typically reach their adult height already by the age of 14 or 15 years while boys might still grow at the age of 16 years. Since the mean age of the teenagers studied in this data is 16.2 years, it is clear that the age affects more to the height of the boys than the girls.

To summarize, there was nothing very surprising in this data experiment except for the fact that nearly all the relationships were very weak. It could potentially be because ordinal data is perhaps not so well-suited for estimating these types of dependence or that some information is lost while editing the data into fully numeric. Nonetheless, we noticed here how important it is to take all the variables into account because, if the gender of the teenagers is not considered, the data suggest that there is some connection between the height and exercising.

5.3 Air pollution in London

Our third and final real data experiment is about different air pollutants. Namely, there is much data available about the air quality, and the relationships related to it are likely stronger than in our previous experiment but not so predictable as in our

first experiment. The data used here is from the website Londonair [33] provided by the Environmental Research Group of Imperial College London.

Air pollution is a very significant public health issue, especially in big cities such as London. Because of this, London Air Quality Network (LAQN) was formed in 1993 to coordinate air quality monitoring in different parts of London and South East England. These measurements are not only used to model air pollution in the whole area and predict its future development but they can also help local authorities with their decision making in topics related to air pollution. [32]

The data set of my third experiment consists of six variables whose values are measured in the monitoring site in Greenwich, Woolwich Flyover during the time period from January 1st, 2019, to January 1st, 2020 [33]. Greenwich is an old London borough but not so densely populated as most of the other boroughs in Greater London, and Woolwich is one of the districts within Greenwich. The monitoring site is near a relatively busy overpass which might go under construction in the near future because of several traffic accidents [10].

The six pollutants measured on this monitoring site include nitric oxide, nitrogen dioxide, oxides of nitrogen, ozone, and PM10 and PM2.5 particulate, which are also the variables we consider here. All of their values are measured in micrograms per cubic meter of air (ug/m^3). Note here that both nitric oxide and nitrogen dioxide are oxides of nitrogen but there are also other gases consisting of nitrogen and oxygen, such as nitrogen monoxide, included in this group of gases. Furthermore, PM10 particulate means particles with diameter less than 10 micrometers and PM2.5 is similarly defined, so a PM2.5 particle is always also a PM10 particle. Thus, we could expect from this that the values of third variable are increasing with respect to the two first ones and the fourth variable is increasing with respect of the fifth one.

By plotting the histograms of these variables, we can see that they are approximately normally distributed but are still all positively skewed. Because of this and to make our third experiment comparable with the two previous ones, we consider once again Spearman's correlation coefficient instead of Pearson's. The values of this coefficient and the other measures of dependence, including the maximal correlation, the mutual information, the informal coefficient of correlation r_1 and the MIC, can be seen in Table 7.

From Table 7, we see that the relationships in this data are much stronger than those in our earlier data experiments. The values of nitric oxide, nitrogen dioxide and oxides of nitrogen are clearly positively correlated with each other, and so are PM10 and PM2.5 particles. There is also noticeable positive correlation between these three first variables and the last mentioned ones but, interestingly, the amount of ozone is negatively correlated with all the other air pollutants.

It can also be computed that while the values of oxides of nitrogen are highly correlated with both nitric oxide and nitrogen dioxide with Spearman's correlation coefficients of 0.985 and 0.911, respectively, the correlation between oxides of nitrogen and the sum of the first variables is greater than this: $r_s = 0.9985$. However, because this coefficient is strictly less than its maximal value 1, we can see that the third variable is not equivalent to the sum of the first two. In fact, from our data, we can actually check that all the variable formed out by decreasing by the sum of nitric oxide and nitrogen dioxide from oxides of nitrogen has only positive values,

Pollutants	r_s	$\rho_{\max}$	MI	r_1	MIC
NO; NO2	0.833	0.833	0.620	0.843	0.541
NO; NOX	0.985	0.993	1.715	0.984	0.918
NO; O3	-0.644	0.657	0.340	0.703	0.311
NO; PM10	0.439	0.481	0.166	0.532	0.171
NO; PM2.5	0.514	0.537	0.235	0.613	0.227
NO2; NOX	0.911	0.914	0.896	0.913	0.646
NO2; O3	-0.584	0.591	0.247	0.624	0.251
NO2; PM10	0.580	0.592	0.234	0.612	0.233
NO2; PM2.5	0.663	0.667	0.326	0.692	0.308
NOX; O3	-0.655	0.665	0.339	0.702	0.319
NOX; PM10	0.502	0.523	0.193	0.566	0.204
NOX; PM2.5	0.584	0.593	0.270	0.646	0.268
O3; PM10	-0.332	0.398	0.110	0.445	0.137
O3; PM2.5	-0.492	0.565	0.218	0.595	0.223
PM10; PM2.5	0.727	0.743	0.420	0.754	0.354

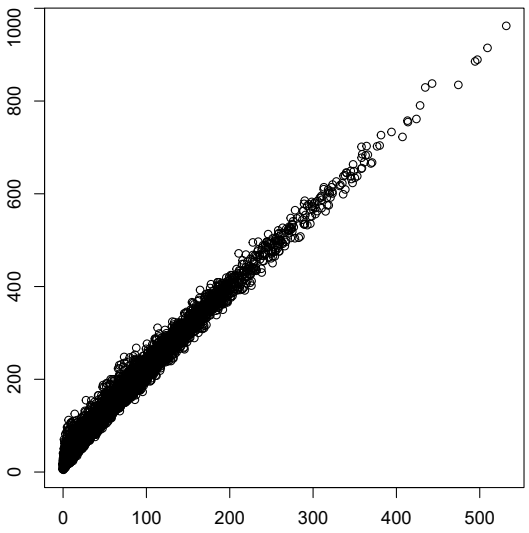
Table 7: Spearman's correlation coefficient r_s, the maximal correlation coefficient $\rho_{\max}$, the mutual information (MI), the informal coefficient of correlation r_1 and the MIC for different pollutants including nitric oxide (NO), nitrogen dioxide (NO2), oxides of nitrogen (NOX), ozone (O3), and PM10 and PM2.5 particulate.

excluding one observation that is most likely an error, and has an average value of 34.0 ug/m^3.

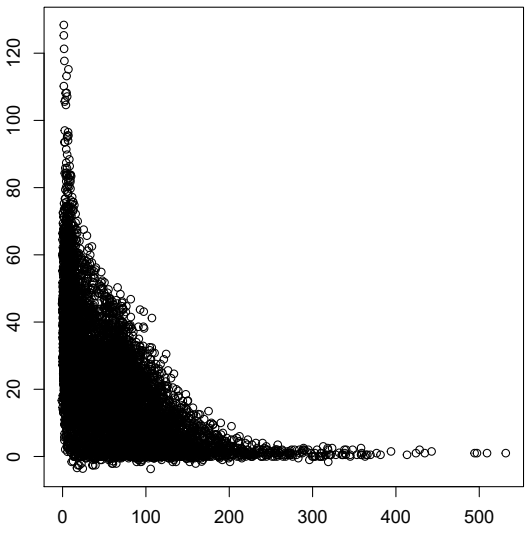
Consider next the negative correlation between ozone and the other air pollutants. While stratospheric ozone has an important task of protecting us from the ultraviolet radiation, ozone at ground level forms environmentally harmful smog and is thus considered an air pollutant. Ozone is formed in the chemical reactions between oxides of nitrogen and volatile organic compounds where oxide molecules are destroyed and freed oxygen atoms connect to each other in sets of three [49]. Since the same process creating ozone at ground level also decreases nitric oxide, nitrogen dioxide and other oxides of nitrogen, we have a clear explanation why these three variables are negative correlated with ozone.

Because the negative correlation between ozone and particulate matter is weaker than both the positive correlation between oxides of nitrogen and particle matter and the negative correlation between oxides of nitrogen and ozone, it is possible that this connection between ozone and particle matter is at least partially explained by the connections from these variables to oxides of nitrogen. Possible other relevant factors are also the temperature and the amount of rain. However, while these weather measurements are surely studied somewhere near the monitoring site of Woolwich Flyover, they are not included in our data.

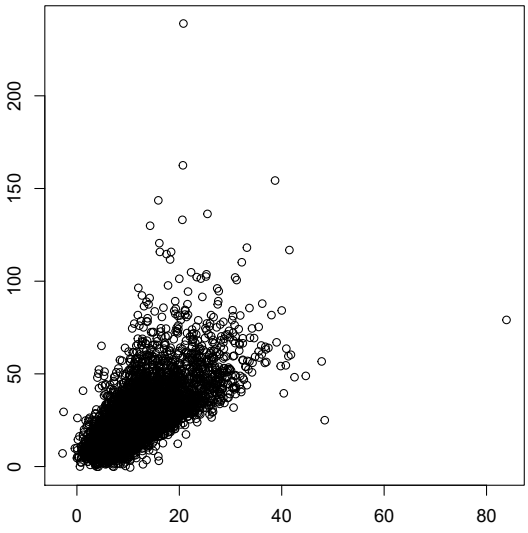
Still, we can study the effect of the temperature on these variables by considering their values against time. We namely notice that most of values of nitric oxide are around 0-100 ug/m^3 during each month of the year 2019, but there are values over 400 ug/m^3 in only January, February and December. Consequently, cold winter weather seems to increase the amount of nitric oxide and similar observations can



(a) Oxidixes of nitrogen against nitric oxide



(b) Ozone against nitric oxide



(c) PM10 against PM2.5 particulate

Figure 18: Scatter plots between a few different variables in the air pollution data.

also be made from three other variables including nitrogen dioxide, oxides of nitrogen and PM10 particulate. The values of ozone peaks up during mid-April instead and the amount of PM2.5 particles does not seem to be affected by the season.

However, the average temperature of the month does not explain the connection between particulate matter and the other pollutants. For instance, Spearman's correlation coefficient has a value of 0.647 between oxides of nitrogen and PM10 particulate during the first 744 observations from the month of January, while we can see from Table 7 that the corresponding number for the whole data is 0.502. Thus, the dependence between these two variables cannot only be a consequence of the fact that they both happen to have higher values during the same months.

Furthermore, we can notice from Table 7 that all the relationships between the variables considered are monotonic. Namely, the differences between the absolute

values of Spearman's correlation coefficient and the maximal correlation are very small. The fact that the values of r_1 are also similar to those of the maximal correlation suggest that there are no non-functional relationships, either. The values of the MIC are small but that was to be expected on based on our first two data experiments and explained by the amount of the statistical noise. One can also verify these assumptions about monotonic and functional relationships by drawing the different scatter plots for these variables as is done in Figure 18 and, in this way, we also notice that several of these relationships are even linear.

To conclude, there are strong linear relationships between different air pollutants. Most of these variables are positively correlated but higher values of ozone occur together with smaller values of the other pollutants and vice versa, likely because of the chemical process behind the formation of this gas. Furthermore, while the season seems to affect the values of most of the air pollutants considered here, it is alone not enough to explain these relationships.

6 Conclusions

In this work, we studied different measures of dependence, including Pearson's, Spearman's and Kendall's correlation coefficients, the maximal correlation coefficient, the mutual information and the maximal information coefficient (MIC) through simulations and a few real data experiments. These coefficients have been designed separately for different purposes during the time period from the late-19th century to the year 2011 and thus have some distinct properties. Here, we summarize and discuss our earlier observations about these measures of dependence.

Firstly, Pearson's, Spearman's and Kendall's correlation coefficients worked in a very expected way. Pearson's coefficient is a very natural choice for studying dependence if it is linear and the variables follow normal distribution. If these assumptions do not hold, Spearman's and Kendall's coefficients might be better options because they recognize monotonic dependence. Kendall's coefficient is also less sensitive to error than the other two coefficients. However, even if the relationship in the data is monotonic but not linear, even Pearson's correlation coefficient can be used to check if this dependence is increasing or decreasing. In simulations, it was also noticed that all these three coefficients have very small absolute values in the case of no dependence and, especially, Pearson's coefficient is not very sensitive to the statistical noise.

In nearly all simulations and data experiments, the maximal correlation coefficient had values greater than the other quantities, indicating that this coefficient finds different types of dependence most effectively. As can be expected, its value decreases as the amount of statistical noise increases but, as noted in our simulations, the amount of this decrease is quite reasonable. Furthermore, the value of the maximal correlation can be computed very fast with the R-functions introduced in Subsection 3.1.

While the value of the maximal correlation itself does not give any sign if the found dependence is decreasing, increasing or non-monotonic, this problem can be solved by checking the value of either Pearson's, Spearman's or Kendall's correlation coefficient. Alternatively, if there are a lot of variables, we can use the maximal correlation to find such pairs that clearly depend on each other and then draw scatter plots to see what kind of dependence there is. Namely, according to the earlier real data experiments, the relationships with high values of the maximal correlation can also be easily recognized visually.

We can also check if the dependence is monotonic or linear by comparing the value of the maximal correlation coefficient to the absolute value of the first three correlation coefficients. Namely, as noted before when studying the data about air pollution, the difference between Spearman's coefficient and the maximal correlation is very small if the dependence is monotonic. From our simulation results, we can also verify that if this difference is close to its greatest possible value, 1, in those cases where the dependence is functional but non-monotonic. Similarly, by focusing on the difference between Pearson's coefficient and the maximal correlation, we can obtain information about whether the dependence is linear or not.

However, one of the disadvantages of the maximal correlation is that it is not designed for non-functional relationships. For instance, we noticed in our earlier

simulations that the values of the information coefficient of correlation exceeds those of the maximal correlation in the case of the cross-shaped dependence when there is even little noise. Still, it is noteworthy that the maximal correlation coefficient have greater values than the MIC that is specifically created to find these types of dependence.

Another issue with the maximal correlation is the question whether it is trustworthy in those cases where the data is very unevenly distributed and focused on a specific point. Namely, it was noted on the first data experiment that the values of this coefficient were very large between the wind direction and the wind speed just because there are several observations were both of these variables have values of 0 in the data. Although Pearson's correlation coefficient has an assumption about normally distributed data, this requirement is not specified for the maximal correlation. As a potential solution, I would suggest computing the value of the maximal correlation by first maximizing the amount of linear correlation as usual and then applying Spearman's correlation coefficient instead of Pearson's.

Mutual information is interesting as a theoretical concept but, as mentioned in the literature, its interpretation is difficult. It cannot be directly compared with the other coefficients and, as can be seen in the results from our simulations, it gives also very different values for each type of dependence even in the case where there is no noise. Still, we can see that the mutual information had values smaller than both the maximal correlation and the MIC in the first simulation with data from two independent variables.

To be able to compare the values of the mutual information, we use the information coefficient of correlation r_1. However, the definition of this coefficient is perhaps not the best one. The values of r_1 increase very fast for small values of the mutual information and the values of r_1 are therefore very large, for instance, in the first simulation model with no dependence between the two variables. On the other hand, modifying this definition might not work very well: Already now the values of r_1 are relatively small when compared to those of the maximal correlation in the simulations, and re-defining r_1 as a more gradual function of the mutual information would only worsen this issue.

Still, because mutual information can find non-functional relationships, it could be used together with the maximal correlation to find connected variables out of several options. Based on our simulation about the cross-shaped dependence and other experiments, it seems that the value of r_1 is larger than the maximal correlation only in the case of non-functional dependence, which could be useful information. However, the opposite being true does not mean that the dependence is functional: We noticed that there are at least some non-functional types of dependence, such as the circle-shaped dependence of our simulations, for which the maximal correlation has larger value than the information coefficient of correlation.

While one can compute the mutual information with R in a very simple and fast way, the obtained results are not necessarily very accurate. The data needs to be discretized first and, according to one of my earlier observations, the number of bins used in this discretization needs to be greater than the default number. This is at least case if the number of observations is not very large, a few thousands or so. In fact, finding the optimal number of the bins here could be an interesting question

for future research.

One of our topics in this work was also applying neural networks to compute the mutual information. The MINE algorithm introduced earlier suits well for this purpose: It recognized dependence from each type of relationship used in the testing and, if there are at least 30000 observations, the outputs of this algorithm converge very consistently during the first few thousand epochs. While one needs to estimate the distributions of the variables before using this method, this is not a difficult task if there are enough observations. Consequently, in order to compute the mutual information, the MINE algorithm is a good choice if there are several thousand observations, and otherwise using the R-functions based on discretization might work better.

Compared with the maximal correlation and the information coefficient of correlation, it was quite surprising how poorly the MIC worked in both simulations and real data experiments here. The presence of statistical noise in the data studied is a known issue with the MIC and it was also noticed numerous times in this work. While the MIC recognizes different relationships very well in noiseless data, even tiny amounts of noise weaken the performance of this coefficient considerably. Furthermore, computing the value of the MIC with R is notably slower than finding the values of the other measures of dependence considered here, which must be taken into account when dealing with a data set consisting of several thousands observations.

The type of the data also seems to affect the value of the MIC. By comparing the results of the simulations and real data experiments, we can notice that the values of the MIC are at smallest for the youth risk behavior data consisting out of ordinal variables that were turned numerical by using mostly integer values. Especially, the value of the MIC was nearly ten times smaller between the age and height of teenagers than it was on average for data consisting out of two independent, normally distributed variables data in our simulations. While the relationships are very weak in this youth risk behavior data also when measured by the other coefficients, the values of Spearman's correlation and the maximal correlation are still larger for the variable pair of the age and height than they are for two independent variables. Consequently, this suggests that the MIC is not designed for data with integer variables.

One of the interesting properties of the MIC is its possible equitability: This coefficient should give similar values with equally noisy relationships regardless of the exact type of dependence. While I noticed clear differences, for instance, between the cubic and sinusoidal relationships, there were also some signs of this equitability property working, too. However, this property is not very useful when the statistical noise has so significant impact on the values of the MIC. To put it simply, it does not help that some coefficient works similarly in different situations if it always performs so inadequately that it cannot be properly utilized.

Thus, all our measures of dependence suit slightly different objectives so it depends on the current situation which one of them should be used. Sometimes, most information about the relationship between variables can be obtained through a combination of these coefficients: For instance, if we need to find dependence effectively either simulated or real data, the values of Spearman's correlation coefficient,

the maximal correlation coefficient and the information coefficient of correlation tell us if there is dependence, if the dependence is increasing, decreasing or non-monotonic, and also if there is non-functional dependence instead. Because of this, understanding the unique features and differences of these coefficients is important.

Index

References

[1] S. Asoodeh, F. Alajaji and T. Linder, On Maximal Correlation, Mutual Information and Data Privacy. *IEEE 14th Canadian Workshop on Information Theory (CWIT)* (2015), 27-31.

[2] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville and R. D. Hjelm, Mutual Information Neural Estimation. *Proceedings of the 35th International Conference on Machine Learning, PMLR, 80* (2018), 531-540.

[3] C. B. Bell, Mutual Information and Maximal Correlation as Measures of Dependence. *Ann. Math. Statist., 33*, 2 (1962), 587-595.

[4] F. Bijma, M. Jonker and A. van der Vaart, *An Introduction to Mathematical Statistics.* Amsterdam University Press, 2017.

[5] T. Bouwmans, S. Javed, M. Sultana and S. K. Jung, Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation. arXiv: 1811.05255v1, (2018).

[6] L. Breiman and J. H. Friedman, Estimating Optimal Transformation for Multiple Regression and Correlation. *J. Am. Stat. Assoc., 80*, 391 (1985), 580-598.

[7] D. Cao, Y. Chen, J. Chen, H. Zhang and Z. Yuan, An improved algorithm for the maximal information coefficient and its application. *R. Soc. Open sci., 8*, 201424 (2021), 1-12.

[8] Centers for Disease Control and Prevention, Youth Risk Behavior Surveillance System (YRBSS), retrieved on June 19th, 2021, from `https://www.cdc.gov/healthyyouth/data/yrbs/index.htm`

[9] M. Çetinkaya-Rundel, D. Diez, A. Bray, A. Y. Kim, B. Baumer, C. Ismay, N. Paterno and C. Barr, Package 'openintro'. R-package version 2.1.0, (2021). `https://cran.r-project.org/web/packages/openintro/openintro.pdf`

[10] D. Chamberlain, Greenwich's notorious Angerstein roundabout could be ripped out, TfL says. *853*, (January 6th, 2020), retrieved on June 25th, 2021, from `https://853.london/2020/01/06/greenwichs-notorious-angerstein-roundabout-could-be-ripped-out-tfl-says/`

[11] C. Chan, A. Al-Bashabsheh, H. P. Huang, M. Lim, D. S. H. Tam and C. Zhao, Neural Entropic Estimation: A faster path to mutual information estimation. arXiv: 1905.12957v2, (2019).

[12] Y. Chen, Y. Zeng, F. Luo and Z. Yuan, New Algorithm to Optimize Maximal Information Coefficient. *PLoS ONE, 11*, 6 (2016), 1-13.

[13] K. Choi and S. Lee, Regularized Mutual Information Neural Estimation. arXiv: 2011.07932v1, (2020).

[14] M. M. Christiansen and K. R. Duffy, Guesswork, Large Deviations, and Shannon Entropy. *IEEE Trans. Inf. Theory, 59*, 2 (2013), 796-802.

[15] D. Curran-Everett, Explorations in statistics: correlation. *Adv Physiol Educ, 34* (2010), 186-191.

[16] B. Efron and T. Hastie, *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science.* Cambridge University Press, 2016. Corrected March 5, 2021

[17] T. van Erven and P. Harremoës, Rényi Divergence and Kullback–Leibler Divergence. *IEEE Trans. Inf. Theory, 60*, 7 (2014), 3797-3820.

[18] J. D. Fehribach, *Multivariable and Vector Calculus.* De Gruyter, 2020.

[19] E. C. Fieller, H. O. Hartley and E. S. Pearson, Tests for Rank Correlation Coefficients. I. *Biometrika, 44*, 3/4 (1957), 470-481.

[20] M. Filosi, R. Visintainer, D. Albanese, S. Riccadonna, G. Jurman and C. Furlanello, Package 'minerva'. R-package version 1.5.8, (2019). `https://cran.r-project.org/web/packages/minerva/minerva.pdf`

[21] S. Fritsch, F. Guenther, M. N. Wright, M. Suling and S. M. Mueller, Package 'neuralnet'. R-package version 1.44.2, (2019). `https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf`

[22] F. Giannini, V. Laveglia, A. Rossi, D. Zanca and A. Zugarini, Neural Networks for Beginners. A fast implementation in Matlab, Torch, TensorFlow. arXiv: 1703.05298v2, (2017).

[23] B. Gierlichs, L. Batina, P. Tuyls and B. Preneel, Mutual Information Analysis. A Generic Side-Channel Distinguisher. *Cryptographic Hardware and Embedded Systems - CHES 2008.* E. Oswald and P. Rohatgi (Eds.) Lecture Notes in Computer Science, 5154 (2008), 426-444.

[24] Y. Goldberg, *Neural Network Methods for Natural Language Processing.* Morgan & Claypool, 2017.

[25] G. Grimmett and D. Welsh, *Probability: An Introduction.* Oxford University Press, 1986. 2nd Ed., 2014.

[26] J. Hauke and T. Kossowski, Comparison of values of Pearson's and Spearman's correlation coefficient on the same sets of data. *Quaestiones Geographicae, 30*, 2 (2011), 87-93.

[27] Ilmatieteenlaitos, data set collected in Nuorgam, Utsjoki, during the time period from June 1st, 2020, to May, 31st, 2021, downloaded on June 3rd, 2021, from `https://en.ilmatieteenlaitos.fi/download-observations`

[28] J. B. Kinney and G. S. Atwal, Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences, 111*, 9 (2014), 3354–3359.

[29] S. Kullback and R. A. Leibler, On information and sufficiency. *Ann. Math. Statist., 22*, 1 (1951), 79-86.

[30] X. Lin, I. Sur, S. A. Nastase, A. Divakaran, U. Hasson and M. R. Amer, Data-Efficient Mutual Information Neural Estimator. arXiv: 1905.03319v2, (2019).

[31] E. H. Linfoot, An Informational Measure of Correlation. *Inf. Control, 1* (1957), 85-89.

[32] Londonair, About londonair, retrieved on June 25th, 2021, from `https://www.londonair.org.uk/LondonAir/General/about.aspx`

[33] Londonair, data set collected in Woolwich Flyover, Greenwich, during the time period from January 1st, 2019, to January, 1st, 2020, downloaded on June 23rd, 2021, from `https://www.londonair.org.uk/london/asp/datasite.asp?site=GR8`

[34] G. Lu, New refinements of Jensen's inequality and entropy upper bounds. *J. Math. Inequalities, 12*, 2 (2018), 403-421.

[35] A. Luedtke and L. Tran, The Generalized Mean Information Coefficient. arXiv: 1308.5712v1, (2013).

[36] B. Macukow, Neural Networks – State of Art, Brief History, Basic Models and Architecture. *15th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM)* (2016) 3-14.

[37] P. E. Meyer, Package 'infotheo'. R-package version 1.2.0, (2015). `https://cran.r-project.org/web/packages/infotheo/infotheo.pdf`

[38] V. Pacelli and M. Azzollini, An Artificial Neural Network Approach for Credit Risk Management. *Journal of Intelligent Learning Systems and Applications, 3* (2011), 103-112.

[39] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher and P. C. Sabeti, Detecting Novel Associations in Large Data Sets. *Science, 334*, 6062 (2011), 1518-1524.

[40] D. N. Reshef, Y. A. Reshef, M. Mitzenmacher and P. C. Sabeti, Equitability Analysis of the Maximal Information Coefficient, with Comparisons. arXiv: 1301.6314v2, (2013).

[41] D. N. Reshef, Y. A. Reshef, H. K. Finucane, P. C. Sabeti and M. Mitzenmacher, Measuring Dependence Powerfully and Equitably. *J. Mach. Learn. Res., 17*, (2016), 1-63.

[42] O. Rioul, This is IT: A Primer on Shannon's Entropy and Information. In *Progress in Mathematical Physics*, Birkhäuser, Springer Nature 2021 (to appear), 49-86.

[43] N. Simon and R. Tibshirani, Comment on "Detecting Novel Associations In Large Data Sets" by Reshef Et Al, Science Dec 16, 2011. arXiv: 1401.7645v1, (2014).

[44] V. P. Singh, *Entropy Theory and Its Application in Environmental and Water Engineering.* John Wiley & Sons, Inc., 2013.

[45] J. Song and S. Ermon, Understanding the Limitations of Variational Mutual Information Estimators. arXiv: 1910.06222v2, (2019).

[46] P. Spector, J. Friedman, R. Tibshirani, T. Lumley, S. Garbett and J. Baron, Package 'acepack'. R-package version 1.4.1, (2016). `https://cran.r-project.org/web/packages/acepack/acepack.pdf`

[47] T. Speed, A Correlation for the 21st Century. *Science, 334*, 6062 (2011), 1502-1503.

[48] G. J. Székely, M. L. Rizzo and N. K. Bakirov, Measuring and testing dependence by correlation of distances. *Ann. Statist., 35* 6 (2007), 2769-2794.

[49] United States Environmental Protection Agency, Ground-level Ozone Pollution, retrieved on June 25th, 2021, from `https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics`

[50] N. Veyrat-Charvillon and F.-X. Standaert, Mutual Information Analysis: How, When and Why? *Cryptographic Hardware and Embedded Systems - CHES 2009.* C. Clavier and K. Gaj (Eds.) Lecture Notes in Computer Science, 5747. (2009), 429-443.

[51] C. Wang, L. Gong, Q. Yu, X. Li, Y. Xie, and X. Zhou, GDLAU: A Scalable Deep Learning Accelerator Unit on FPGA. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 36*, 3 (2017), 513-517.

[52] C. Xiao, J. Ye, R. M. Esteves and C. Rong, Using Spearman's correlation coefficients for exploratory data analysis on big dataset. *Concurrency Computat.: Pract. Exper, 28* (2016), 3866-3878.

[53] Y. Xu, Y. Xu, Q. Qian, H. Li and R. Jin, Towards Understanding Label Smoothing. arXiv: 2006.11653, (2020).

[54] M. Yamada, MINE: Mutual Information Neural Estimation in pytorch. PyTorch code, (2018). `https://github.com/MasanoriYamada/Mine_pytorch`

[55] Y. Zhang, S. Jia, H. Huang, J. Qiu and C. Zhou, A Novel Algorithm for the Precise Calculation of the Maximal Information Coefficient. *Sci Rep, 4* (2014), 1-5.

[56] J. Zupan, Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them. *Acta Chimica Slovenica, 41*, 3 (1994), 327-352.

R and PyTorch codes

Here, the reader can find some of the most important R and PyTorch codes used in this work.

R code for Section 3

The following R code shows how data is generated from the models (3.1)-(3.9), how Figures 7, 8 and 11 are plotted, and how the results of Table 3 are obtained.

```
#R code for 3
#ormrai 2021-05-03
#FILE: tilg101.R begins

#libraries
library(acepack)
library(infotheo)
library(minerva)

#function for generating data from the models (3.1)-(3.9)
simxy<-function(j,n,sigma){
  #Independent variables
  if(j==1){
    x<-rnorm(n,0,sqrt(0.1))
    y<-rnorm(n,0,sqrt(0.1))
  }
  #Linear dependence
  if(j==2){
    x<-rnorm(n,0,sqrt(0.1))
    y<-x+rnorm(n,0,sigma)
  }
  #Cubic dependence
  if(j==3){
    x<-rnorm(n,0,sqrt(0.1))
    y<-x^3+1/3*x+rnorm(n,0,sigma)
  }
  #Quadratic dependence
  if(j==4){
    x<-rnorm(n,0,sqrt(0.1))
    y<-3*x^2-1+rnorm(n,0,sigma)
  }
  #Sinusoidal dependence
  if(j==5){
    x<-rnorm(n,0,sqrt(0.1))
    y<-sin(9*x)+rnorm(n,0,sigma)
  }
  #Cross-shaped dependence
```

```
  if(j==6){
    k<-rbinom(1,n,0.5)
    x<-c(rnorm(k,0,sigma/2),rnorm(n-k,0,sqrt(0.1)))
    y<-c(rnorm(k,0,sqrt(0.1)),rnorm(n-k,0,sigma/2))
  }
  #Circular dependence
  if(j==7){
    k<-rnorm(n,0,1)
    l<-rnorm(n,1,sigma)
    x<--l*cos(k)
    y<--l*sin(k)
  }
  #Two functions
  if(j==8){
    k<-rnorm(n,0,1)
    for(u in 1:n){
      if(k[u]>=0){
        x[u]<-2*k[u]/3-1
        y[u]<-x[u]+rnorm(1,0,sigma)
      }else{
        x[u]<--(-k[u])^0.1+0.1
        y[u]<--(x[u]-0.1)^10+1+rnorm(1,0,sigma)
      }
    }
  }
  #Checkerboard dependence
  if(j==9){
    x<-rnorm(n,0,sqrt(0.1))
    y<-rnorm(n,0,sqrt(0.1))
    for(u in 1:n){
      while(((floor(3*x[u])-floor(3*y[u]))%%2)==1){
        x[u]<-rnorm(1,0,0.1)
        y[u]<-rnorm(1,0,0.1)
      }
    }
    y<-y+rnorm(n,0,sigma/3)
  }
  dxy<-cbind(x,y)
  return(dxy)
}

#plotting Figure 7
n<-1000
sigma<-0
par(mfrow=c(3,3))
for(j in 1:9){
```

```
  par(pty='s')
  dxy<-simxy(j,n,sigma)
  plot(dxy,xlim=c(-1,1),ylim=c(-1,1),xaxt='n',yaxt='n',
  ann=FALSE)
}

#studying the impact of sigma
n<-1000
s<-seq(0,0.3,by=0.03)
df1<-as.data.frame(matrix(NA,1,7))
names(df1)<-c("sigma","p","s","k","MCC","r1","MIC")
for(j in 1:length(s)){
  sigma<-s[j]
  df<-as.data.frame(matrix(NA,1,6))
  names(df)<-c("p","s","k","mcc","r1","MIC")
  for(i in 1:1000){
    dxy<-simxy(2,n,sigma)
    #2 means the linear model above, can be replaced!
    x<-dxy[,1]
    y<-dxy[,2]
    df[i,1]<-cor(x,y)
    df[i,2]<-cor(x,y,method="spearman")
    df[i,3]<-cor(x,y,method="kendall")
    fxy<-ace(x,y)
    df[i,4]<-cor(fxy$tx,fxy$ty)
    disc<-discretize(data.frame(x,y))
    mi<-mutinformation(disc$x,disc$y)
    df[i,5]<-sqrt(1-exp(-2*mi))
    df[i,6]<-mine(x,y)$MIC
  }
  df1[j,1]<-sigma
  for(i in 1:6){
    df1[j,i+1]<-mean(df[,i])
  }
  print(sigma)
}
print(df1)

#plotting Figure 8
xl<-c(0,0.3)
yl<-c(min(df1[,2:7]),1)
par(mfrow=c(1,1))
par(pty='m')
plot(df1[,1],df1[,2],type="l",xlim=xl,
      ylim=yl,ylab="",xlab=expression(sigma),
      lty=1,lwd=1,cex.lab=1.3,cex.axis=1.1)
```

```
par(new=TRUE)
plot(df1[,1],df1[,3],type="l",xlim=xl,ylim=yl,
     axes=FALSE,xlab="",ylab="",lty=2,lwd=1)
par(new=TRUE)
plot(df1[,1],df1[,4],type="l",xlim=xl,ylim=yl,
     axes=FALSE,xlab="",ylab="",lty=3,lwd=2)
par(new=TRUE)
plot(df1[,1],df1[,5],type="l",xlim=xl,ylim=yl,
     axes=FALSE,xlab="",ylab="",lty=4,lwd=1)
par(new=TRUE)
plot(df1[,1],df1[,6],type="l",xlim=xl,ylim=yl,
     axes=FALSE,xlab="",ylab="",lty=5,lwd=1)
par(new=TRUE)
plot(df1[,1],df1[,7],type="l",xlim=xl,ylim=yl,
     axes=FALSE,xlab="",ylab="",lty=6,lwd=1)
points(df1[,1],df1[,2],pch=1)
points(df1[,1],df1[,3],pch=4)
points(df1[,1],df1[,4],pch=5)
points(df1[,1],df1[,5],pch=3)
points(df1[,1],df1[,6],pch=6)
points(df1[,1],df1[,7],pch=8)
legend("topright",
       legend=c("r","rs",expression(tau),expression(rho),
       "r1","MIC"),pch=c(1,4,NA,3,6,8),lty=c(1,2,3,4,5,6),
       lwd=c(1,1,2,1,1,1),cex=1.3)
#add triangle to the legend, i.e. points(0.2675,0.89,pch=5)

#studying the equitability of the MIC
n<-1000
k<-1000
df<-as.data.frame(matrix(NA,k,8))
for(i in 1:k){
  sigma<-abs(rnorm(1,0,1))
  dxy<-simxy(2,n,sigma)
  x<-dxy[,1]
  y<-dxy[,2]
  df[i,1]<-cor(x,y)^2
  df[i,2]<-mine(x,y)$MIC
  sigma<-2/3*sigma
  dxy<-simxy(3,n,sigma)
  x<-dxy[,1]
  y<-dxy[,2]
  df[i,3]<-cor(y,x^3+1/3*x)^2
  df[i,4]<-mine(x,y)$MIC
  sigma<-2*sigma
  dxy<-simxy(4,n,sigma)
```

```
  x<-dxy[,1]
  y<-dxy[,2]
  df[i,5]<-cor(y,3*x^2-1)^2
  df[i,6]<-mine(x,y)$MIC
  sigma<-1.65*sigma
  dxy<-simxy(5,n,sigma)
  x<-dxy[,1]
  y<-dxy[,2]
  df[i,7]<-cor(y,sin(9*x))^2
  df[i,8]<-mine(x,y)$MIC
  if(i%%100==0){
    print(i)
  }
}

#plotting Figure 11
par(pty='s')
plot(1,type="n",xlim=c(0,1),ylim=c(0,1),pch=3,
     ylab="MIC",xlab="R^2",cex.lab=1.3,cex.axis=1.1)
for(i in 1:k){
  points(df[i,1],df[i,2],pch=3,col="blue")
  points(df[i,3],df[i,4],pch=3,col="steelblue1")
  points(df[i,5],df[i,6],pch=3,col="darkblue")
  points(df[i,7],df[i,8],pch=3)
}
legend("topleft",
       legend=c("linear","cubic","quadratic",
       "sinusoidal"),pch=c(16,16,16,16),
       col=c("blue","steelblue1","darkblue",
       "black"),cex=1.3)

#studying the impact of the number of observations (Table 3)
sigma<-0.1
n1<-c(5,10,30,100,300,500,700,1000,3000)
df1<-as.data.frame(matrix(NA,1,8))
names(df1)<-c("n","p","s","k","MCC","MI","r1","MIC")
for(j in 1:length(n1)){
  n<-n1[j]
  df<-as.data.frame(matrix(NA,1,7))
  names(df)<-c("p","s","k","mcc","mi","r1","MIC")
  for(i in 1:1000){
    dxy<-simxy(3,n,sigma)
    x<-dxy[,1]
    y<-dxy[,2]
    df[i,1]<-cor(x,y)
    df[i,2]<-cor(x,y,method="spearman")
```

```
    df[i,3]<-cor(x,y,method="kendall")
    fxy<-ace(x,y)
    df[i,4]<-cor(fxy$tx,fxy$ty)
    disc<-discretize(data.frame(x,y))
    mi<-mutinformation(disc$x,disc$y)
    df[i,5]<-mi
    df[i,6]<-sqrt(1-exp(-2*mi))
    df[i,7]<-mine(x,y)$MIC
  }
  df1[j,1]<-n1[j]
  for(i in 1:7){
    df1[j,i+1]<-mean(df[,i])
  }
  print(n)
}
print(df1)

#FILE: tilg101.R ends
```

PyTorch code for Subsection 4.3

The following PyTorch code is the implementation of the MINE algorithm use in the simulations of Subsection 4.3.

```
#PyTorch codes for 4.3
#ormrai 2021-05-30
#FILE: tilg0.ipynb begins

import torch
from torch.autograd import Variable
import torch.nn as nn
import torch.nn.functional as F
from tqdm import tqdm
import holoviews as hv
import bokeh
hv.extension('bokeh')
import numpy as np
import pandas as pd

# data
# fix j to either 1 (no dependence), 2 (linear), 3 (cubic),
# 4 (quadratic) or 5 (sinusoidal)
j = 1
sigma = 0.3
def gen_x():
```

```
  return np.random.normal(0.,np.sqrt(0.1),[data_size,1])

def gen_y(x):
  return func(x)+np.random.normal(0.,sigma,[data_size,1])

if j==1:
  def gen_y(x):
    return np.sign(np.random.normal(0.,np.sqrt(0.1),
    [data_size,1]))

if j==2:
  def func(x):
    return x

if j==3:
  def func(x):
    return x**3 + x / 3

if j==4:
  def func(x):
    return 3*x**2 - 1

if j==5:
  def func(x):
    return np.sin(9 * x)

data_size = 30000
x=gen_x()
y=gen_y(x)

H=10
n_epoch = 1000

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(1, H)
        self.fc2 = nn.Linear(1, H)
        self.fc3 = nn.Linear(H, 1)

    def forward(self, x, y):
        h1 = F.relu(self.fc1(x)+self.fc2(y))
        h2 = self.fc3(h1)
        return h2

model = Net()
```

```
optimizer = torch.optim.Adam(model.parameters(), lr=0.01)
plot_loss = []
for epoch in tqdm(range(n_epoch)):
    x_sample=gen_x()
    y_sample=gen_y(x_sample)
    y_shuffle=np.random.permutation(y_sample)

    x_sample = Variable(torch.from_numpy(
    x_sample).type(torch.FloatTensor), requires_grad = True)

    y_sample = Variable(torch.from_numpy(
    y_sample).type(torch.FloatTensor), requires_grad = True)

    y_shuffle = Variable(torch.from_numpy(
    y_shuffle).type(torch.FloatTensor),
    requires_grad = True)

    pred_xy = model(x_sample, y_sample)
    pred_x_y = model(x_sample, y_shuffle)

    ret = torch.mean(pred_xy) - torch.log(torch.mean(
    torch.exp(pred_x_y)))

    loss = - ret  # maximize
    plot_loss.append(loss.data.numpy())
    model.zero_grad()
    loss.backward()
    optimizer.step()


plot_x = np.arange(len(plot_loss))
plot_y = np.array(plot_loss).reshape(-1,)

#printing results
y100 =-plot_y[(len(plot_loss)-101):(len(plot_loss)-1)]
mi_est = np.mean(y100)
results = {'object': ['j','n','sigma','mi_est'],
        'value': [j, data_size, sigma, mi_est]
        }
df = pd.DataFrame (results, columns=['object','value'])
print('|')
print(df)
print(mi_est)

#saving the array in a text file
file = open("f1.txt", "w+")
```

```
content = str(-plot_y)
content = content.replace('[', '')
content = content.replace(']', '')
file.write(content)
file.close()

hv.Curve((plot_x, -plot_y))

#FILE: tilg0.ipynb ends
```

R code for Subsection 5.1

The following R code can be used in the real experiments of Subsection 5.1 if Nuorgam's weather data from Ilmatieteenlaitos is imported to R with the name *itld*.

```
#R code for 5.1
#ormrai 2021-06-05
#FILE: tilg501.R begins

#libraries
library(acepack)
library(infotheo)
library(minerva)

#creating a dataframe with 8 variables:
#1.air pressure, 2.relative humidity, 3.snow, 4.temperature,
#5.dew point, 6.wind direction, 7.gust speed, 8.wind speed
f<-function(u){
  as.numeric(as.vector(u)[2:8762])
}
df<-cbind(f(itld$V7),f(itld$V9),f(itld$V11),f(itld$V12),
          f(itld$V13),f(itld$V15),f(itld$V16),f(itld$V17))
df<-df[complete.cases(df),]

#checking this dataframe
dim(df)
summary(df)
head(df)

#plotting histograms
for(i in 1:8){
  hist(df[,i])
}
hist(df[,6],breaks=100,main="Histogram of wind direction",
```

```
      xlab="Wind direction in degrees",col="white",
      cex.axis=1.1,cex.lab=1.3,border="blue")

#finding the mode of wind direction
getmode<-function(v){
  uniqv<-unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
}
getmode(df[,6])

#plotting a few variables against each other
par(pty='s')
#temperature vs snow
plot(df[,4],df[,3],cex.axis=1.1)
#temperature vs dew
plot(df[,4],df[,5],cex.axis=1.1)
#wind direction vs gust speed
plot(df[,6],df[,7],cex.axis=1.1)
#wind speed vs gust speed
plot(df[,8],df[,7],cex.axis=1.1)

#computing measures of dependence from the data
x<-c(rep(1,7),rep(2,6),rep(3,5),rep(4,4),rep(5,3),6,6,7)
y<-c(2:8,3:8,4:8,5:8,6:8,7,8,8)
colxy<-cbind(x,y,rep(0,28),rep(0,28),rep(0,28),rep(0,28),
    rep(0,28))
for(i in 1:28){
  colxy[i,3]<-cor(df[,colxy[i,1]],df[,colxy[i,2]],
    method='spearman')
  fxy<-ace(df[,colxy[i,1]],df[,colxy[i,2]])
  colxy[i,4]<-cor(fxy$tx,fxy$ty)
  disc<-discretize(data.frame(df[,colxy[i,1]],
    df[,colxy[i,2]]))
  mi<-mutinformation(disc[,1],disc[,2])
  colxy[i,5]<-mi
  colxy[i,6]<-sqrt(1-exp(-2*mi))
  colxy[i,7]<-mine(df[,colxy[i,1]],df[,colxy[i,2]])$MIC
}
df1<-round(colxy,digits=3)

#printing the results for a table in latex
for(i in 1:28){
  print(c("&",df1[i,3],"&",df1[i,4],"&",df1[i,5],
    "&",df1[i,6],"&",df1[i,7],"\\"),quote=FALSE)
}
```

```
#studying the maximal correlation between the wind direction
#and the two types of wind speeds

#mcc for wind direction vs gust speed
fxy<-ace(df[,6],df[,7])
print(cor(fxy$tx,fxy$ty))

#mcc for wind direction vs wind speed
fxy<-ace(df[,6],df[,8])
print(cor(fxy$tx,fxy$ty))

#mcc for wind direction vs gust speed with Spearman
fxy<-ace(df[,6],df[,7])
print(cor(fxy$tx,fxy$ty,method="spearman"))

#mcc for wind direction vs wind speed with Spearman
fxy<-ace(df[,6],df[,8])
print(cor(fxy$tx,fxy$ty,method="spearman"))

#mcc for wind direction vs gust speed without zero
#observations of the gust speed
sdf<-subset(df,df[,7]>0)
dim(sdf)
fxy<-ace(sdf[,6],sdf[,7])
print(cor(fxy$tx,fxy$ty))

#mcc for wind direction vs wind speed without zero
#observations of the gust speed
fxy<-ace(sdf[,6],sdf[,8])
print(cor(fxy$tx,fxy$ty))

#FILE: tilg501.R ends
```