0.1 Medidas de correlação

Francisco Aparecido Rodrigues
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
francisco@icmc.usp.br

0.1.1 Correlação de Pearson

Quando temos duas variáveis, é importante analisamos como essas variáveis se relacionam linearmente. Uma medida muito comum para medirmos a correlação entre variáveis é o coeficiente Pearson, que é definido por:

$$\rho_{X,Y} = \frac{\mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y]}{\sqrt{\mathrm{E}[X^2] - [\mathrm{E}[X]]^2}\,\sqrt{\mathrm{E}[Y^2] - [\mathrm{E}[Y]]^2}} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}, \tag{1}$$

onde $\mathrm{cov}(X,Y)$ é a covariância entre X e Y. Na prática, usamos a definição para uma amostra de dados:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X) \cdot \mathrm{var}(Y)}}. \tag{2}$$

A partir dessa equação, podemos fornecer uma interpretação dessa medida. A operação no numerador $(\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}))$ visa centralizar os dados em uma mesma média, igual a zero. Notem que a média de $\sum_{i=1}^{n}(x_i - \bar{x})$ é igual zero, pois:

$$\sum_{i=1}^{n}(x_i - \bar{x}) = E(X - E(X)) = E(X) - E(X) = 0,$$

onde usamos $E(c) = \sum_i cP(X = x_i) = c\sum_i P(X = x_i) = c$, onde c é uma constante. Ou seja, inicialmente, centralizamos os dados em uma mesma média. Essa parte da equação é igual à covariância, que é positiva se as variáveis X e Y têm a mesma tendência — se X aumenta (diminui), Y aumenta (diminui). Caso a tendência seja contrária, isto é, se X aumenta (diminui) então Y diminui (aumenta), a covariância é negativa. Caso não haja relação entre as variáveis, a covariância é nula. No entanto, a covariância não oferece um valor limitante positivo ou negativo, dependendo da escala dos dados. Notem que a covariância tem o mesmo sinal que o coeficiente de Pearson, mas não é normalizada. Para termos um valor definido em um intervalo, dividimos a covariância pelo produto do desvio padrão das variáveis X e Y. Ou seja, essa divisão ajusta a escala dos dados, de modo que ambos tenha a mesma importância em termos da correlação. Assim, o coeficiente de correlação é definido no intervalo $-1 \leq \rho \leq 1$, sendo $\rho = 1$ se X e Y forem totalmente correlacionados, ou $\rho = -1$ se forem anti-correlacionados. Na figura 1 mostramos alguns exemplos de relações entre X e Y e os respectivos coeficientes de correlação.

Abaixo, mostramos o código em Python para gerar um conjunto de dados e calcular a correlação.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import pearsonr

N = 100 # numero de elementos nos vetores
```

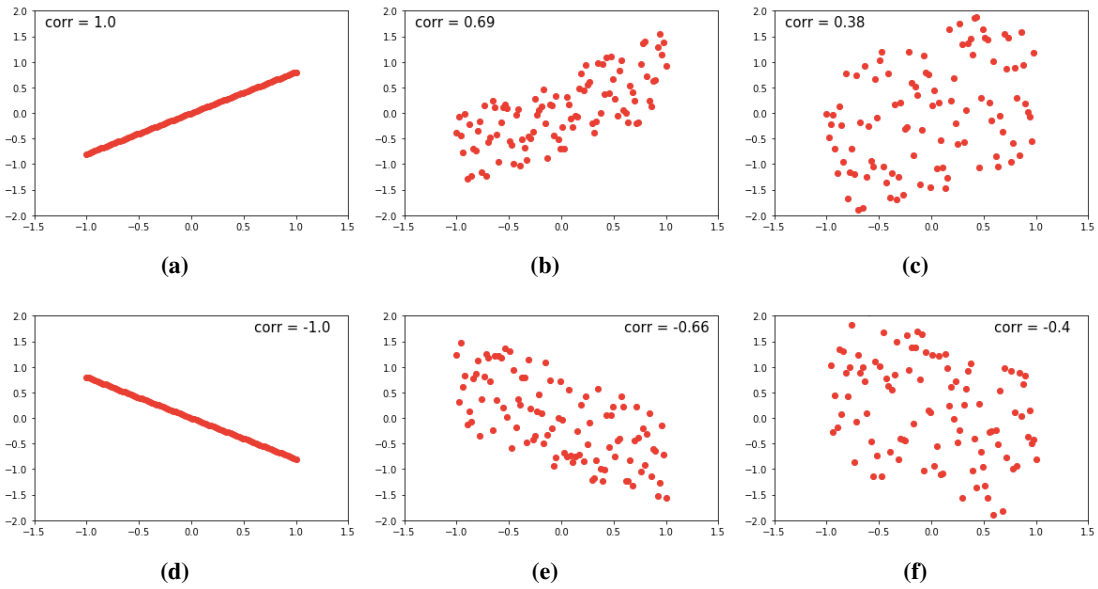

Figura 1: Exemplos de relações entre X e Y e os respectivos coeficientes de correlação.

```
X = np.linspace(-1,1, N)
erro = np.random.uniform(-1,1,N)
for sigma in np.arange(0,2,0.2):
    Y = -0.8*X + erro*sigma # troque o sinal de menos por mais para corr. positiva.
    plt.plot(X,Y, 'ro')
    plt.xlim(-1.5,1.5)
    plt.ylim(-2, 2)
    corr, p_value = pearsonr(X, Y) # calcula o coeficiente de correlacao de Pearson
    corr = int(corr*100)/100 # mostra apenas duas casas decimais
    string = 'corr = '+ str(corr)
    plt.text(0.6,1.7, string, fontsize=15) # posicao do valor da corr. no grafico
    plt.show(True)
```

O valor *p* retornado na função pode ser interpretado como a probabilidade de observar uma alta correlação (ou correlação negativa) na amostra se a correlação verdadeira for nula. Há diversas formas de analisar a significância da correlação e deve-se atentar para os métodos existentes, que podem levar a resultados diferentes.

A correlação de Pearson é imporante para analisar a relação entre as variáveis. Se duas variáveis são altamente correlacionadas, é adequado remover uma delas, de modo a diminuir a redundância nos dados. Por exemplo, na figura 2, mostramos a correlação entre os atributos da base de atributos da flor Iris. Por meio dessa matriz, podemos selecionar as variáveis menos correlacionadas, de modo a diminuir a redundância. Apesar de bastante simples, há métodos mais efetivos para essa tarefa, como os algoritmos para selecionar os atributos mais relevantes em apredizado supervisionado.

Para obtermos a figura 2, usamos os comandos em Python:

```
import pandas as pd
data = pd.read_csv('data/iris.csv', header=(0))

corr = data.corr()
plt.figure(figsize=(7, 5))
```

```
plt.imshow(corr, cmap='Blues', interpolation='none', aspect='auto')
plt.colorbar()
plt.xticks(range(len(corr)), corr.columns, rotation='vertical')
plt.yticks(range(len(corr)), corr.columns);
plt.suptitle('Correlation between variables', fontsize=15, fontweight='bold')
plt.grid(False)
plt.show()
```

Um observação importante sobre a medida de correlação é que ela não implica independência entre as variáveis. Para que duas variáveis X e Y sejam independentes, deve-se verificar a probabilidade condicional. Se X é independente de Y, então temos que $P(X|Y) = P(Y)$. Ou seja,

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} \Rightarrow P(X,Y) = P(X)P(Y).$$

Desse modo, para que duas variáveis sejam independente, a distribuição conjunta deve ser igual ao produto das marginais (no caso contínuo: $f(x,y) = f_X(x)f_Y(y)$) [1]. O fato da correlação ser nula implica que as variáveis aleatórias são não correlacionadas, mas não que elas são indepentes. No entanto, se as variáveis forem independentes, temos que elas também são não correlacionadas, pois no caso contínuo (também vale para o discreto),

$$E(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf(x,y)dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_X(x)f_Y(y)dxdy = \int_{-\infty}^{\infty} xf_X(x)dx \int_{-\infty}^{\infty} yf_Y(y)dy = E[X]E[Y].$$

Logo, quando inserimos esse resultado na equação 1, vemos que a correlação será nula se X e Y forem independentes.

0.1.2 Correlação de Spearman

Além da correlação de Pearson, é possível quantificar a relação entre variáveis por meio do coeficiente de *correlação de Spearman* e da *informação mútua*. O coeficiente de Spearman é similar ao coeficiente de Pearson, mas ao invés de considerar o valor observado da variável, é considerada a sua ordem nos dados. Ou seja, se tivermos um vetor com valores $\{9,2,3,5\}$, usaremos $\{4,1,2,3\}$. Assim, a correlação de Spearman é igual ao coeficiente de Pearson aplicado aos valores da ordem de duas variáveis. Essa medida avalia a relação monotônica entre duas variáveis contínuas ou ordinais e não é sensível a assimetrias na distribuição, nem à presença de outliers, já que consideramos a ordem e não os valores da variáveis. Em uma relação monotônica, as variáveis tendem a mudar juntas mas não necessariamente a uma taxa constante. Ou seja, a correlação de Spearman mede a intensidade da relação entre variáveis ordinais, sendo uma medida não paramétrica. Notem que enquanto a correlação de Pearson mede relações lineares, a de Spearman mede apenas relações monotônicas (lineares ou não-lineares).

Na figura 3 mostramos a diferença entre os coeficiente de Pearson e Spearman. Notem que o coeficiente de Spearman é próximo de um, pois a função logaritmíca é monotônica (sempre crescente).

Em Python, o código para comparar os coeficientes de correlação é dado a seguir.

```
import numpy as np
import matplotlib.pyplot as plt
```

[1] No caso discreto, a marginal de uma variável aleatória X é $P(X = x) = \sum_y P(X = x, Y = y)$, e no caso contínuo $P(a \leq X \leq b) = \int_a^b f_X(x)dx$, onde $f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy$, sendo $f(x,y)$ a distribuição de probabilidade conjunta de (X,Y).

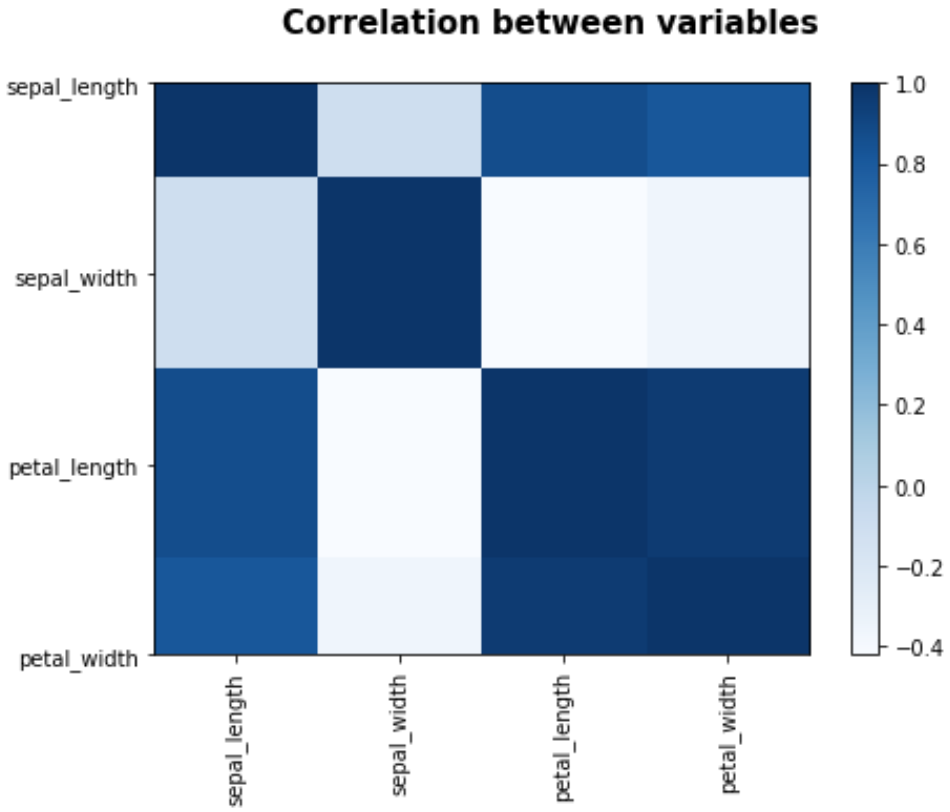

Figura 2: Matriz de correlação entre os atributos da flor Iris. A intesidade das cores representa o nível de correlação entre as variáveis.

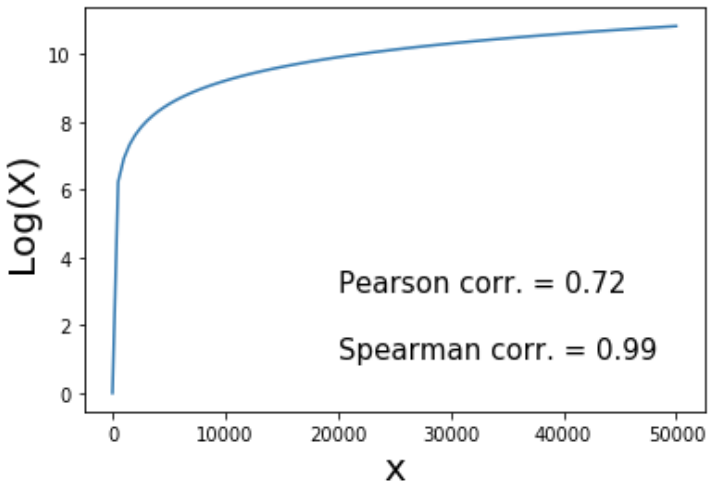

Figura 3: Comparação dos coeficientes de Pearson e Spearman para a função logarítimica.

```
from scipy.stats import pearsonr, spearmanr

N = 100
x = np.linspace(1, 50000, N)
z = np.log(x)

plt.plot(x, z)
plt.xlabel("x")
plt.ylabel("Log(X)")
corr, p_value = pearsonr(x, z) # correlacao de Pearson
corrs, p_values = spearmanr(x, z) # correlacao de Spearman
corr = int(corr*100)/100 # mostra apenas duas casas decimais
corrs = int(corrs*100)/100
string = 'Pearson corr. = '+ str(corr)
plt.text(20000,3, string, fontsize=15)
string = 'Spearman corr. = '+ str(corrs)
plt.text(20000,1, string, fontsize=15)
plt.show()
```

0.1.3 Informação mútua

Conforme discutimos anteriormente, a correlação de Pearson igual a zero não implica em independência entre as variáveis. Uma maneira de quantificarmos a relação de dependência entre variáveis é considerar Teoria da Informação, introduzida por Claude Shannon em 1948 [1]. A quantidade de informação recebida ao observar uma variável aleatória discreta X com distribuição $P(X = x) = p(x)$ pode ser definida como a quantidade de incerteza sobre o seu valor antes de realizarmos o experimento. Ou seja, a observação de uma variável aleatória X é mais informativa se o seu valor é mais difícil de se prever *a priori*, baseando-se apenas em $p(x)$. Em outras palavas, quanto mais "difícil" é a advinhação do valor de X antes do experimento, maior a sua quantidade informação associada.

A soma do ganho de informação ao observar dois eventos x e y, que possuem as variáveis aleatórias X e Y associadas, respectivamente, deve ser igual ao ganho quando observados seperadamente, isto é,

$$h(x,y) = h(x) + h(y), \tag{3}$$

se X e Y forem variáveis independentes,

$$P(X = x, Y = y) = p(x,y) = p(x)p(y), \tag{4}$$

onde $p(x)$ é a distribuição de probabilidade marginal da variável aleatória X. Para que essas duas relações sejam válidas, temos que $h(x)$ deve ser uma função logarítmica do inverso de $p(x)$, pois quanto maior a chance de acerto, menor a quantida de informação contída na variável. Logo,

$$h(x) = \log_2 \frac{1}{p(x)} = -\log_2 p(x), \tag{5}$$

onde $h(x) \geq 0$. A quantidade média de informação transmitida em um processo é dada pela esperança de

$h(x)$,

$$H[x] = -\sum_x h(x)p(x) = -\sum_x p(x)\log_2 p(x), \tag{6}$$

que é denominada *entropia de Shannon*.

Vamos considerar um exemplo. Em um dado com seis faces, a entropia associada:

$$H[x] = -\sum_{x=1}^{6} \frac{1}{6}\log_2 \frac{1}{6} = 2,58$$

Como usamos a base 2, esse valor está em bits. Se usarmos a base e, teremos o resultado em nats. Nesse livro, quando usarmos a função log sem definir a base, estaremos usando a função logaritmo natural.

De maneira geral, para a distribuição uniforme discreta[2], que retorna a probabilidade de um valor observado em um dado de N lados e é definida por:

$$P(X = x) = \frac{1}{N}, \qquad x = 0, 1, \ldots, N, \tag{7}$$

temos que a entropia de Shannon:

$$H[X] = -\sum_x \frac{1}{N}\log \frac{1}{N} = \log N. \tag{8}$$

Esse é o valor máximo de uma variável aleatória, pois a distribuição uniforme é a menos informativa possível.

Uma implementação de entropia de Shannon em Python para um dado de N faces:

```
from scipy.stats import entropy
N = 6
Px = np.ones(N)*1/N
H = entropy(Px, base=2)
print("Entropia de Shannon: ", H)
```

Para o caso de duas variáveis, podemos definir a entropia conjunta, usando argumentos semelhantes aos apresentados para o caso de uma variável. Assim,

$$H[X,Y] = -\sum_x \sum_y p(x,y)\log p(x,y). \tag{9}$$

Por exemplo, se as variáveis representam $X = \{\text{quente, frio}\}$ e $Y = \{\text{seco,úmido}\}$ e p(quente, seco) = 1/2, p(quente, úmido) = 1/4, p(frio, seco) = 1/4 e p(frio, úmido) = 0. Então

$$H[X,Y] = -\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{4}\log\frac{1}{4} + 0\log 0\right] = \frac{3}{2}.$$

A entropia condicional é definida em termos da distribuição de probabilidade condicional:

$$H[X|Y] = -\sum_x \sum_y p(x,y)\log p(x|y). \tag{10}$$

Para duas variáveis, podemos definir ainda a entropia relativa, ou divergência de Kullback-Leibler.

[2]Distribuição uniforme discreta: $p(x) = 1/N, x \in \{a, a+1, \ldots, b-1, b\}$ e $N = b - a + 1$, a e b inteiros com $b \leq a$.

Vamos supor que $p(x)$ representa a distribuição desconhecida, e que modelamos um conjunto de dados através de outra distribuição $q(x)$. Assim, a divergência de Kullback-Leibler é definida por,

$$D_{KL}(p||q) = \sum_x p(x) \ln \left(\frac{q(x)}{p(x)} \right) dx. \tag{11}$$

O valor retornado por essa medida pode ser interpretado como a perda de informação ao usar $q(x)$ para prever $p(x)$. $D_{KL}(p||q)$ também é usada para comparar distribuições, embora ela não seja uma métrica no sentido matemático, pois não obedece à desigualdade triangular e, em geral, $D_{KL}(p||q) \neq D_{KL}(q||p)$[3]. De fato, $D_{KL}(p||q)$ é uma medida de ineficiência em usar $q(x)$ para aproximar a distribuição verdadeira $p(x)$. Quanto mais próximo de zero, menor é a perda de informação.

Vamos considerar um exemplo. Sejam as distribuições $P(X=0) = 0,25$, $P(X=1) = 0,55$ e $P(X=2) = 0,2$ e $Q(X=0) = Q(X=1) = Q(X=2) = 0,33$. Então,

$$D_{KL}(P||Q) = 0,25 \ln \frac{0,25}{0,33} + 0,55 \ln \frac{0,55}{0,33} + 0,25 \ln \frac{0,2}{0,33} = 0.101$$

$$D_{KL}(Q||P) = 0,33 \ln \frac{0,33}{0,25} + 0,33 \ln \frac{0,33}{0,55} + 0,33 \ln \frac{0,33}{0,2} = 0.099$$

Quanto mais próximo de zero, mais similares são as distribuições. Notem que$D_{KL}(p||q) \leq D_{KL}(q||p)$. Isso se deve ao fato de que $D_{KL}(p||q)$ usa o valor a distribuição de probabilidade $p(x)$ para calcular a esperança, enquanto que $D_{KL}(q||p)$ usa $q(x)$. Se $D_{KL}(q||p) = 0$, temos que as duas distribuições são idênticas.

Como a divergência de Kullback–Leibler pode ser usada para comparar distribuições, podemos usá-la para determinar o nível de independência de duas variáveis aleatórias. Nesse caso, se duas variáveis X e Y são independentes, então $P(X=x, Y=y) = p(x,y) = p(x)p(y)$ (no caso contínuo, a função densidade de probabilidade conjunta $f(x,y) = f_X(x) f_Y(y)$). Assim, para verificamos o nível de independência entre X e Y, usamos:

$$I[X,Y] = D_{KL}(p(x,y)||p(x)p(y)) = -\int\int p(x,y) \ln \left(\frac{p(x)p(y)}{p(x,y)} \right) dxdy. \tag{12}$$

No caso discreto:

$$I[X,Y] = D_{KL}(p(x,y)||p(x)p(y)) = -\sum\sum p(x,y) \ln \left(\frac{p(x)p(y)}{p(x,y)} \right). \tag{13}$$

Essa medida é chamada *informação mútua* entre as variáveis X e Y. Como estamos usando logaritmo natural, a informação mútua é medida em nats.

A informação mútua é uma medida da dependência mútua entre as duas variáveis X e Y. Se $I[X,Y] = 0$ temos que X e Y são independentes. Um alto valor de $I[X,Y]$ é obtido quando é possível prever o valor de X quando Y é observado (ou vice versa). Ou seja, quando maior a dependência de Xe Y, maior é o valor da informação mútua.

[3]Definição de métrica: seja $\mathbb{S}$ um conjunto e d uma função $d : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$. Dizemos que d é uma métrica se: (i) d é positivamente definida, $d(x,y) \geq 0$, $(x,y) \in \mathbb{S}$, (ii) d é simétrica $d(x,y) = d(y,x)$, e (ii) d obedece à desigualdade triangular, $d(x,z) \leq d(x,y) + d(y,z)$ para $x,y,z \in \mathbb{S}$.

Podemos comparar a informação mútua e a correlação. Para a covariância, temos

$$\mathrm{cov}(X,Y) = E[XY] - E[X]E[Y] = \sum_x \sum_y xyp(x,y) - \left(\sum_x xP(x)\right)\left(\sum_y yp(y)\right)$$

Ou seja,

$$\mathrm{cov}(X,Y) = \sum_x \sum_y xy\left[p(x,y) - p(x)p(y)\right]. \tag{14}$$

Já para a informação mútua, usando a equação 0.1.3 na forma discreta,

$$I[X,Y] = \sum_x \sum_y p(x,y)\left[\ln(p(x,y)) - \ln(p(x)p(y))\right]. \tag{15}$$

Desse modo, vemos que a informação mútua não leva em conta se a relação entre X e Y é linear ou não, enquanto que a covariância pode ser igual a zero, mesmo quando X e Y não são independentes. Por outro lado, a covariância pode ser calculada diretamente a partir do conjunto de dados e não requer que as probabilidades sejam conhecidas, já que considera os momentos da distribuição. Como essas duas medidas quantificam diferentes relações entre as variáveis, é importante considerar ambas na descrição de um conjunto de dados. Notem que a principal limitação no uso da informação mútua é a estimação das distribuições de probabilidades.

Para calcular a entropia de Kullback-Leibler em Python, usamos o código abaixo.

```
from scipy.stats import entropy
P = [0.36,0.48,0.16]
Q = [0.33, 0.33, 0.33]
print('KL(P,Q) = ', entropy(P,Q, base = np.exp(1))) #informacao mutua
print('KL(Q,P) = ', entropy(Q,P, base = np.exp(1)))
```

Bibliografia

[1] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.