UNIVERSITÉ DE GENÈVE

ANALYSE ET TRAITEMENT DE L'INFORMATION
14X026

# TP 1: Linear Algebra

*Author:* Joao Costa

*E-mail:* Joao.Costa@etu.unige.ch

November 2022

UNIVERSITÉ
DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

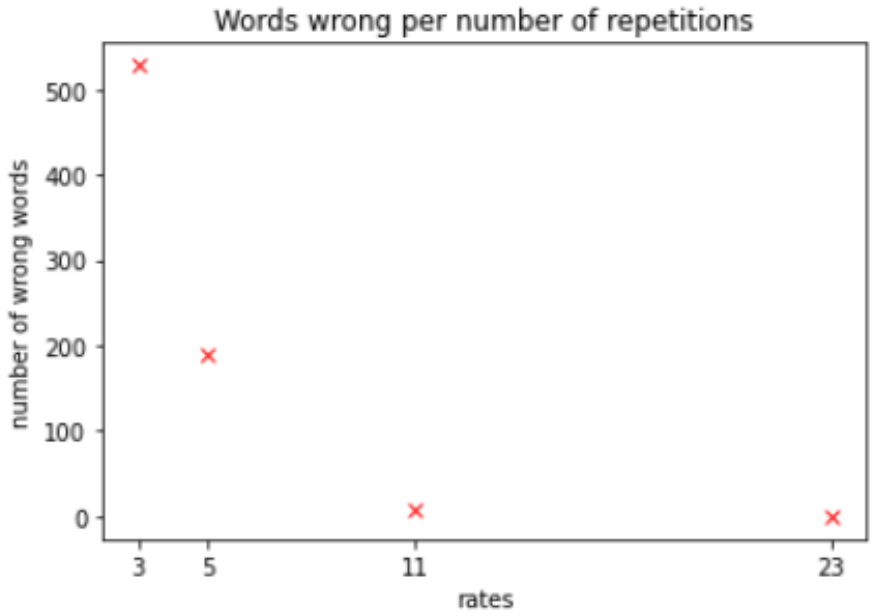
# Exercise 1. Chain rules for probabilities

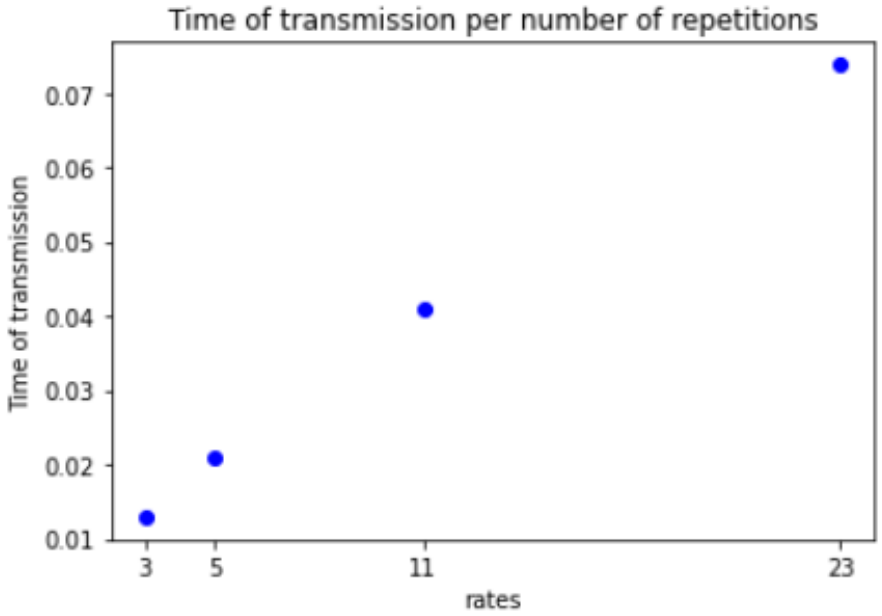
1. 1
   (a) $P(X,Y,Z) = P(X) \cdot p(Y|X) \cdot p(Z|X,Y)$ ->(apply rule)
   (b) $P(X,Z) = P(X) \cdot P(Z|X)$
   (c) $P(X,Y) = P(X) \cdot P(Y|X)$
   (d) $P(Y,Z) = P(Y) \cdot P(Z|X)$
2. 2
   (a) $P(X_1, X_2, ..., X_7) = P(X_1) \cdot P(X_2) \cdot P(X_4) \cdot P(X_5) \cdot P(X_3|X_1, X_2) \cdot P(X_6|X_4, X_5, X_3) \cdot P(X_7|X_6)$
   (b) $P(X_1, X_3, X_5, X_7) = P(X_1) \cdot P(X_3|X_1) \cdot P(X_5) \cdot P(X_7|X_5, X_3)$
   (c) $P(X_2, X_4, X_6, X_7) = P(X_2) \cdot P(X_4) \cdot P(X_6|X_4, X_2) \cdot P(X_7|X_6)$
   (d) $P(X_3, X_6, X_7) = P(X_3) \cdot P(X_6|X_3) \cdot P(X_7|X_6)$
   (e) $P(X_1, X_2, X_4, X_5) = P(X_1) \cdot P(X_2) \cdot P(X_4) \cdot P(X_5)$
3. 3
   (a) $P(X_1, X_2, ..., X_7) = P(X_1) \cdot P(X_2) \cdot P(X_4) \cdot P(X_5|X_2) \cdot P(X_3|X_1, X_2) \cdot P(X_6|X_4, X_5, X_3) \cdot P(X_7|X_6, X_4)$
   (b) $P(X_1, X_3, X_5, X_7) = P(X_1) \cdot P(X_3|X_1) \cdot P(X_5) \cdot P(X_7|X_5, X_3)$
   (c) $P(X_2, X_4, X_6, X_7) = P(X_2) \cdot P(X_4) \cdot P(X_6|X_4, X_2) \cdot P(X_7|X_6, X_4)$
   (d) $P(X_3, X_6, X_7) = P(X_3) \cdot P(X_6|X_3) \cdot P(X_7|X_6)$
   (e) $P(X_1, X_2, X_4, X_5) = P(X_1) \cdot P(X_2) \cdot P(X_4) \cdot P(X_5|X_2)$

## Exercise 2. Communication through noisy channels

See file exo_2.ipynb for code

1. There are on avearage 50% wrong words, which is expected, as each word is composed of 5 bits, so 2 words make up a total of 10 bits, since there is a 10% bit error rate, 1 out of every 2 words should be wrong.
2. As expected the error rate is much lower.
3. By adding repetitions, the error rate becomes lower
4. Obviously by making the message longer, we increase the transfer time

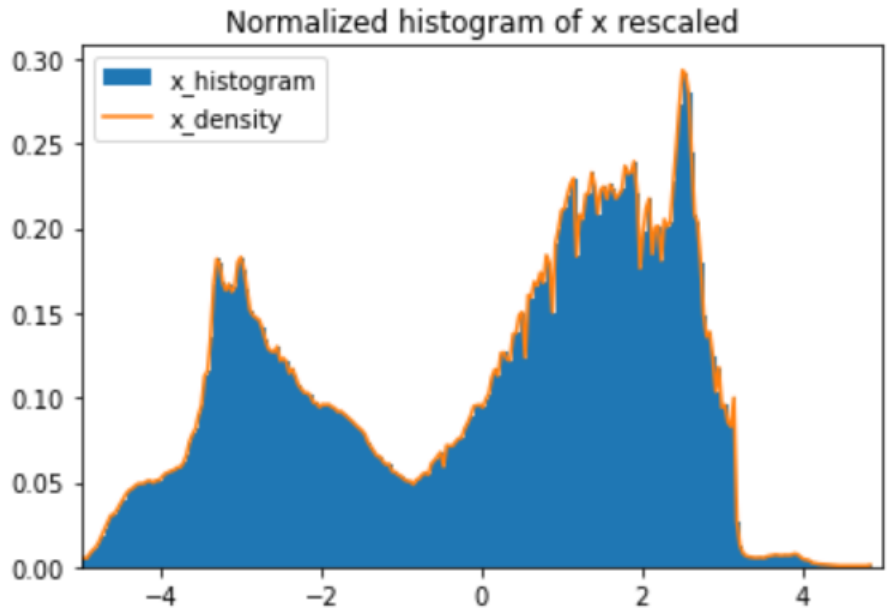


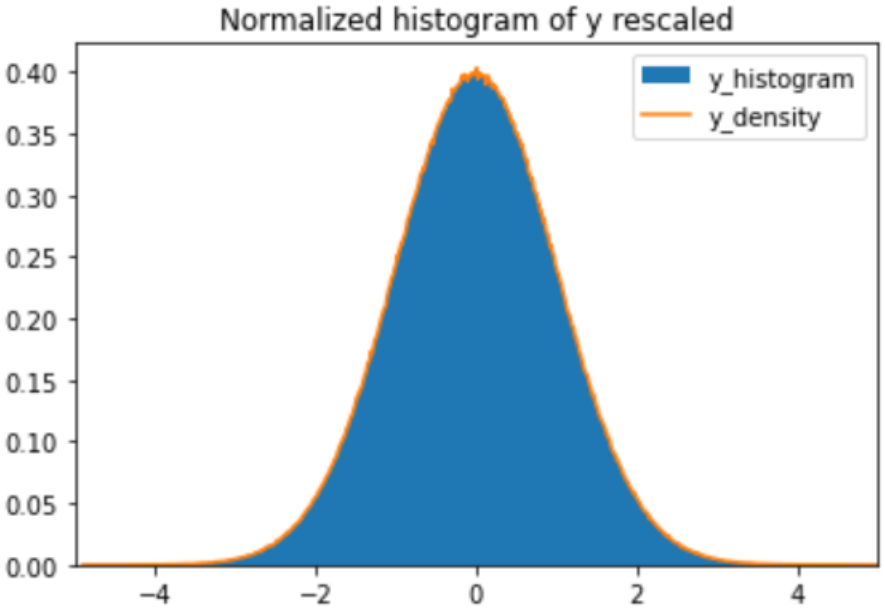



## Exercise 3. Moments

See file exo_3.ipynb for code

1.
```
x = 5 * (x - np.mean(x)) / np.max(np.abs(x - np.mean(x)))
# rescaling between -5 and 5
y = np.random.randn(len(x))
y = y[-5 < y]
y = y[y < 5]
```

2.



3.
```
def mfX(t, x_X, y_X):
    return np.dot(y_X * np.exp(t * x_X[:-1]), x_X[1:] - x_X[:-1])

def mfY(t):
    return integrate.quad(lambda x: np.exp(-x ** 2 / 2) * np.exp(x * t) / np.sqrt(2 * np.pi), -5, 5)[0]
```

4.

```python
def moments(data, n):
    return [np.sum((data) ** k) / len(data) for k in range(n)]

def amf(t, m):
    u = [t ** k / np.math.factorial(k) for k in range(len(m))]
    return np.dot(u, m)
```

```python
n_m = 7
m_X = moments(x, n_m)
print('Moments of X : ', m_X)
m_Y = moments(y, n_m)
print('Moments of Y : ', m_Y)
```

```
Moments of X :  [1.0, 5.48443567728444e-15, 4.969513782353428, -4.623405633388906, 45.058973938097395, -86.27514065057433, 542.
9180664370957]
Moments of Y :  [1.0, 0.0003855931791905327, 1.0000755504445094, 0.0009402074440729503, 2.998895800364789, 0.01214545165438814
6, 14.974178113627703]
```