TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ MÔN XÁC SUẤT THỐNG KÊ**

**ỨNG DỤNG CHO CNTT**

**Applied Probability and Statistics for IT**

*Người hướng dẫn*: **ThS. NGUYỄN LÂM**

*Người thực hiện*: **PHẠM ĐẶNG THANH TRUNG 522H0148**

Lớp **: 22H50302**

Khoá  **: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Trước hết, em muốn bày tỏ lòng biết ơn chân thành đến Trường Đại học Tôn Đức Thắng vì đã bao gồm môn học Xác suất thống kê trong chương trình giảng dạy. Đặc biệt, em muốn gửi lời cảm ơn sâu sắc tới giảng viên bộ môn, Thầy Nguyễn Lâm, đã hướng dẫn và truyền đạt kiến thức hữu ích, giúp em hoàn thành bài tiểu luận.

Dường như kiến thức là vô tận, nhưng việc tiếp thu kiến thức luôn có những hạn chế riêng. Do đó, trong quá trình hoàn thành bài tiểu luận, chắc chắn rằng sẽ không tránh khỏi những sai sót. Tôi rất mong nhận được sự góp ý từ Thầy để bài tiểu luận của em có thể hoàn thiện hơn.

Em xin gửi lời cảm ơn và biết ơn chân thành!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Nguyễn Lâm. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 6 tháng 11 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*TRUNG*

*PHẠM ĐẶNG THANH TRUNG*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Môn học Xác suất và Thống kê là một phân ngành của lĩnh vực Thống kê, tập trung vào việc khám phá khả năng xảy ra của các sự kiện trong một tập dữ liệu. Nó dựa trên các khái niệm về xác suất để đánh giá, mô tả, và phân tích các hiện tượng có tính ngẫu nhiên và không chắc chắn. Môn này đặc biệt quan tâm đến việc nghiên cứu các phân phối xác suất, tính toán xác suất, và đặc điểm của các biến ngẫu nhiên. Xác suất và Thống kê cung cấp các công cụ để đánh giá và dự đoán kết quả trong các bối cảnh không rõ ràng hoặc không thể dự đoán trước. Môn này có ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm kinh tế, y học, khoa học xã hội, công nghệ thông tin và nhiều lĩnh vực nghiên cứu khác, giúp giải thích và hiểu các quy luật của các sự kiện không chắc chắn trong thế giới thực.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc150463247)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc150463248)

[TÓM TẮT iv](#_Toc150463249)

[MỤC LỤC 1](#_Toc150463250)

[CHƯƠNG 1: THƯ VIỆN STATISTICS TRONG PYTHON 3](#_Toc150463251)

[1.1 Hàm statistics.mean(Data) 3](#_Toc150463252)

[1.2 Hàm statistics.fmean(Data, Weights=None) 4](#_Toc150463253)

[1.3 Hàm statistics.median(Data) 5](#_Toc150463254)

[1.4 Hàm geometric\_mean(Data) 6](#_Toc150463255)

[1.5 Hàm statistics.harmonic\_mean(Data, weights=None) 7](#_Toc150463256)

[1.6 Hàm statistics.median\_low(Data) 8](#_Toc150463257)

[1.7 Hàm statistics.median\_high(Data) 9](#_Toc150463258)

[1.8 Hàm statistics.median\_grouped(Data, Interval=1) 10](#_Toc150463259)

[1.9 Hàm statistics.mode(Data) 11](#_Toc150463260)

[1.10 Hàm statistics.multimode(Data) 11](#_Toc150463261)

[1.11 Hàm statistics.quantiles(Data, \*, n=4, method=’exclusive’) 12](#_Toc150463262)

[1.12 Hàm statistics.pstdev(Data, mu=None) 13](#_Toc150463263)

[1.13 Hàm statistics.pvariance(Data, mu=None) 14](#_Toc150463264)

[1.14 Hàm stdev() 15](#_Toc150463265)

[1.15 Hàm statistics.variance(data, xbar=None) 16](#_Toc150463266)

[1.16 Hàm statistics.covariance(x,y,/) 17](#_Toc150463267)

[1.17 Hàm statistics.correlation(x,y,/) 18](#_Toc150463268)

[1.18 Hàm statistics.linear\_regression(x,y,/, \*,proportional=False) 19](#_Toc150463269)

[CHƯƠNG 2: THUẬT TOÁN HISTOGRAM EQUALIZATION 20](#_Toc150463270)

[2.1 Khái niệm và vấn đề: 20](#_Toc150463271)

[2.2 Điều kiện và rằng buộc: 21](#_Toc150463272)

[2.3 Công thức tính Histogram toán học: 21](#_Toc150463273)

[2.4 Công thức tính Histogram equalization toán học: 21](#_Toc150463274)

[2.5 Thuật Toán: 22](#_Toc150463275)

[2.6 Ví dụ: 22](#_Toc150463276)

[2.7 Nhận xét: 24](#_Toc150463277)

[CHƯƠNG 3: THUẬT TOÁN HISTOGRAM MATCHING 24](#_Toc150463278)

[3.1 Khái niệm và vấn đề: 25](#_Toc150463279)

[3.2 Rằng buộc và điều kiện: 25](#_Toc150463280)

[3.3 Công thức toán học: 25](#_Toc150463281)

[3.4 Thuật toán: 26](#_Toc150463282)

[3.5 Ví dụ: 26](#_Toc150463283)

[3.6 Nhận xét: 27](#_Toc150463284)

[CHƯƠNG 4: THỰC HIỆN GIẢI THUẬT 27](#_Toc150463285)

[4.1 HISTOGRAM EQUALIZATION 28](#_Toc150463286)

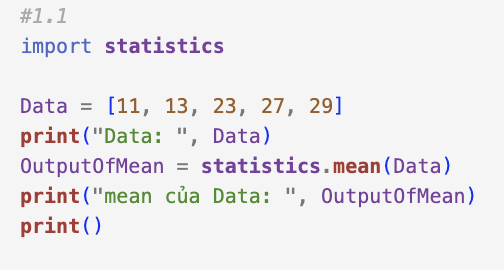
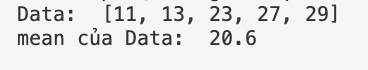
[4.2 HISTOGRAM MATCHING: 32](#_Toc150463287)

[TÀI LIỆU THAM KHẢO 38](#_Toc150463288)

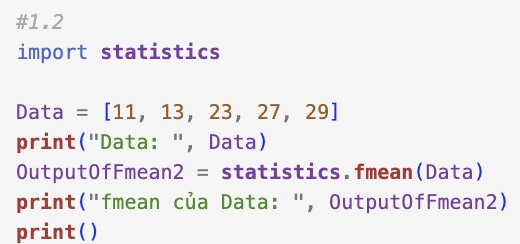
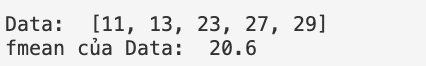
CHƯƠNG 1: THƯ VIỆN STATISTICS TRONG PYTHON

*Tổng quan: Trong phân tích dữ và thống kê, thư viện statistic đóng 1 vai trò rất quan trọng. Nó cung cấp rất nhiều hàm quan trọng, giúp tính toán các giá trị trung tâm, các giá trị phân vị, độ lệch chuẩn và phương sai. Ngoài ra nó còn có thể tính toàn nhiều phép tính rất phức tạp khác nhau. Nhờ thế mà thư viện Statistics được ững dụng vào khoa học dữ liệu và học máy và xử lý dữ liệu trong các nghiên cứu khoa học khác nhau trong cuộc sống. Thư viện cung cấp các hàm cơ bản như mean(), midan(), stdev(), quantiles(), fmean(),…*

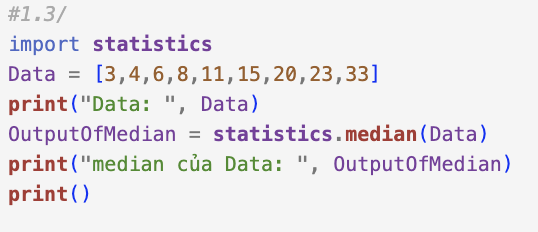
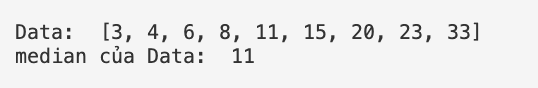
1. Hàm statistics.mean(Data)

* Hàm này dùng để tính giá trị trung bình cộng của 1 tập hợp dữ liệu Data trong python.
* Data là một tập các giá trị đầu vào dùng để tính giá trị trung bình. Data có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm này chỉ nhận một tham số đầu vào, đó là Data.
* Trả về kết quả: Hàm mean() trả về giá trị trung bình cộng của Data. Giá trị trung bình cộng được tính bằng cách cộng tất cả các giá trị trong tập hợp Data rồi chia cho số lượng các giá trị đó rồi trả về kết quả cuối cùng**, giúp giải quyết nhiều bài toán khác nhau.**
* Nếu “data” trống thì **StatisticsError** sẽ xuất hiện.
* CODE:****
* KẾT QUẢ: ****

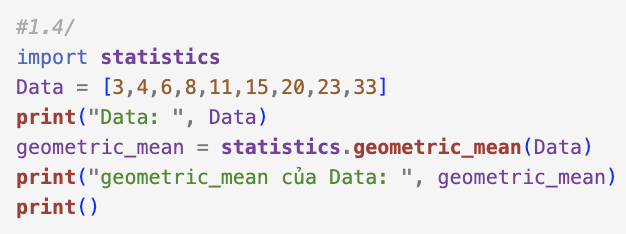
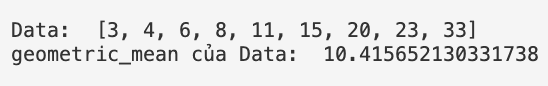
1. Hàm statistics.fmean(Data, Weights=None)

* Hàm được sử dụng để tính giá trị trung bình cộng của Data nhanh hơn và chính xác hơn so với hàm mean() thông thường.
* Data: là một tập các giá trị đầu vào. Data có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Weights: là một danh sách gồm các trọng số tương ứng với các giá trị trong Data. Mỗi giá trị trong weights sẽ được sử dụng để nhân với giá trị tương ứng trong data khi tính toán giá trị trung bình. Nếu không cung cấp tham số này hoặc khai báo nó là “None”, thì tất cả các giá trị trong Data sẽ có trọng số bằng nhau. “Weights” Phải cùng độ dài với “data” hoặc lỗi ValueError sẽ được xuất hiện.
* Tham số đầu vào: hàm này chỉ hai tham số đầu vào, đó là Data và Weights. Chỉ hỗ trờ tử bản python 3.11 trở đi với tham số Weights. Các phiên bản python cũ trước thì chỉ có 1 tham số là Data.
* Kết quả trả về: hàm này trả về giá trị trung bình cộng của Data. Nó trả về rất nhanh và chính xác.
* CODE:
* KẾT QUẢ:

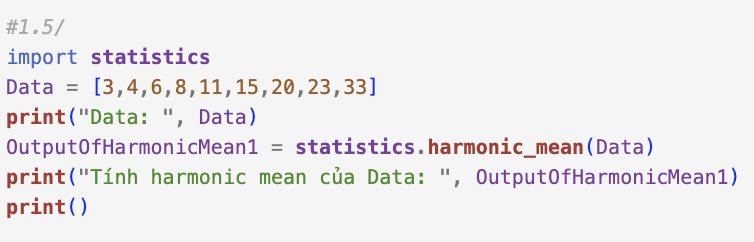
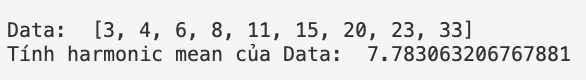
1. Hàm statistics.median(Data)

* Hàm được sử dụng để tính giá trị trung vị của một tập hợp dữ liệu đầu vào là Data. Giá trị trung vị là giá trị nằm ở vị trí giữa của một tập hợp dữ liệu đã được sắp xếp theo thứ tự từ thấp đến cao hoặc từ cao đến thấp. Số lượng phần tử trong tập hợp là lẻ thì giá trị trung vị nằm giữa, ngược lại thì sẽ là trung bình cộng của hai giá trị nằm giữa.
* Data là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple **hoặc** hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm chỉ nhận duy nhất 1 tham số vào là Data để tìm trung vị.
* Kết quả trả về: Hàm trả về giá trị trung vị của Data. Số lượng phần tử trong tập hợp là lẻ thì giá trị trung vị nằm giữa, ngược lại thì sẽ là trung bình cộng của hai giá trị nằm giữa.
* CODE:
* KẾT QUẢ:

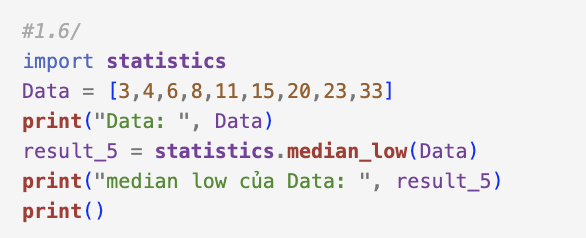
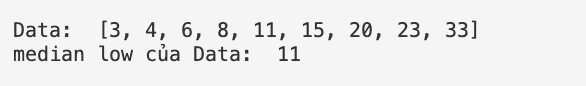
1. Hàm geometric\_mean(Data)

* Hàmđược sử dụng để tính trung bình hình học của một tập hợp dữ liệu Data nhập vào. Trung bình hình học là số trung bình của một dãy các tích số, phép tính.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm chỉ nhận duy nhất 1 tham số vào là Data.
* Kết quả trả về: Hàm này sẽ trả về trung bình hình học của Data đầu vào. Trung bình hình học của n là căn bậc n của tích của tất cả các số đó, với n là số lượng các số.
* CODE:
* KẾT QUẢ:

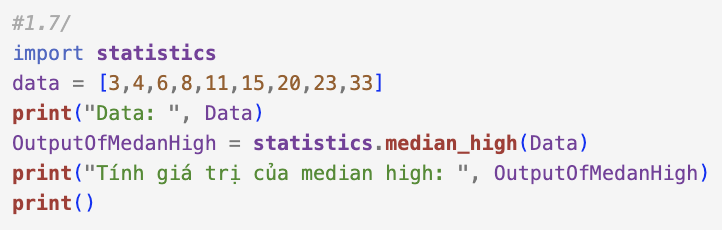
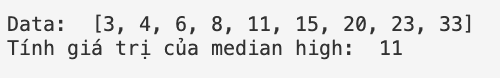
1. Hàm statistics.harmonic\_mean(Data, weights=None)

* Hàmđược sử dụng để tính giá trị trung bình điều hòa của một tập hợp dữ liệu Data. Giá trị trung bình điều hòa là một loại giá trị trung bình mà mối quan hệ nghịch với nhau.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Weights: là một danh sách gồm các trọng số tương ứng với các giá trị trong Data. Mỗi giá trị trong weights sẽ được sử dụng để nhân với giá trị tương ứng trong data khi tính toán giá trị trung bình điều hoà. Nếu không cung cấp tham số này hoặc khai báo nó là “None”, thì tất cả các giá trị trong Data sẽ có trọng số bằng nhau.
* Tham số đầu vào: Gồm 2 tham số là Data và Weights.
* Kết quả trả về: trả về giá trị trung bình hài hoà của Data. Hàm sẽ tính nghịch đảo của mỗi giá trị trong tập hợp Data rồi tính tổng tất cả các nghịch đảo đó. Cuối cùng tiến hành chia số lượng phần tử trong Data cho tổng nghịch đảo.
* CODE:
* KẾT QUẢ:

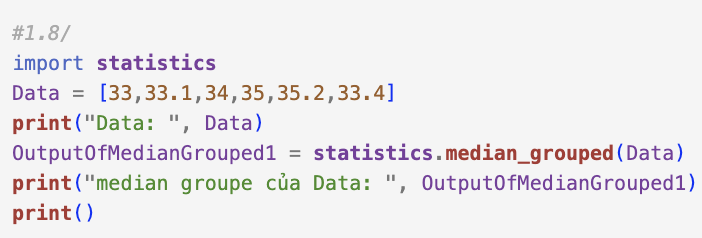
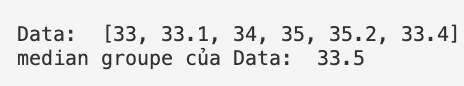
1. Hàm statistics.median\_low(Data)

* Hàm được sử dụng để tính giá trị trung vị thấp của tập hợp dữ liệu Data được nhập vào. Giá trị trung vị thấp là giá trị giữa của tập hợp Data sau khi nó được sắp xếp theo thứ tự tăng dần.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm chỉ nhận duy nhất 1 tham số vào là Data.
* Kết quả trả về: Hàm trả về giá trị trung vị thấp của Data. Trường hợp số lượng phần tử là lẻ, thì giá trị trung vị thấp sẽ là giá trị nằm chính giữa. Ngược lại là chẵn, thì giá trị trung vị thấp sẽ là giá trị thấp hơn trong hai giá trị ở giữa.
* CODE:
* KẾT QUẢ:

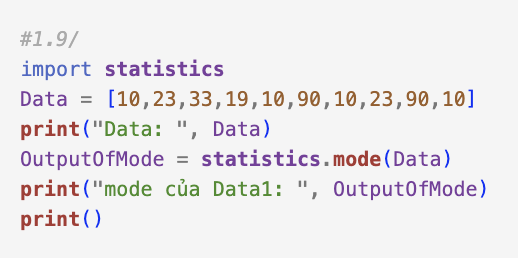
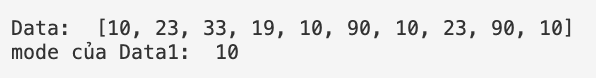
1. Hàm statistics.median\_high(Data)

* Hàm được sử dụng để tính giá trị trung vị cao của tập hợp dữ liệu Data được nhập vào. Giá trị trung vị thấp là giá trị giữa của tập hợp Data sau khi nó được sắp xếp theo thứ tự tăng dần tương tự như tính giá trị trung vị thấp.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm chỉ nhận duy nhất 1 tham số vào là Data.
* Kết quả trả về: Hàm trả về giá trị trung vị cao của Data. Trường hợp số lượng phần tử là lẻ, thì giá trị trung vị cao sẽ là giá trị nằm chính giữa. Ngược lại là chẵn, thì giá trị trung vị thấp sẽ là giá trị cao hơn trong hai giá trị ở giữa.
* CODE:
* KẾT QUẢ:

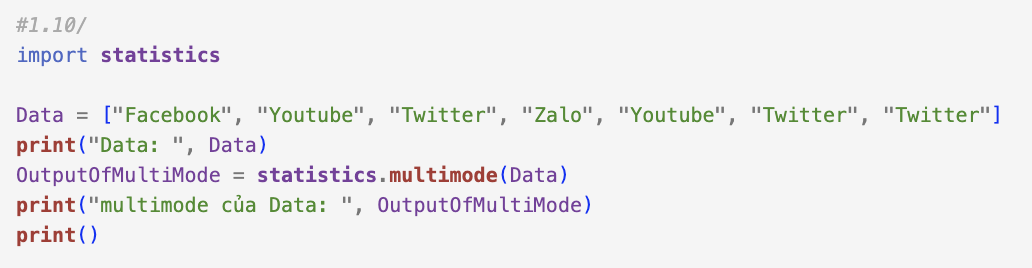
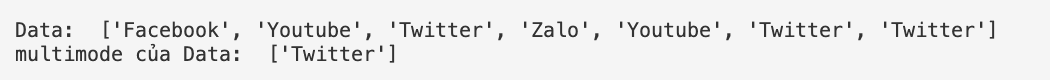
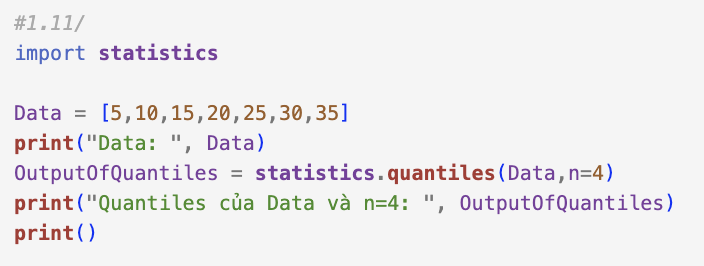
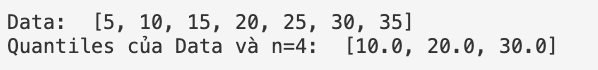
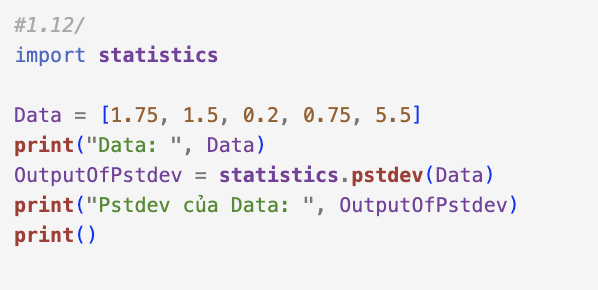
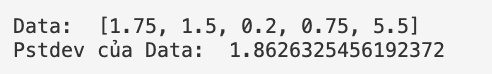
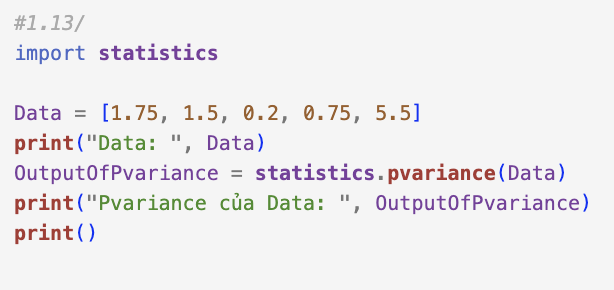
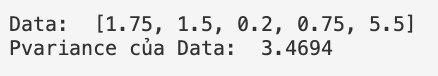
1. Hàm statistics.median\_grouped(Data, Interval=1)

* Hàm được sử dụng để tính trung vị của dữ liệu nhóm Data.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Interval: là tham số đầu vào tuỳ do mình thiết lập, nó là kích thước hoặc là khoảng cách giữa 2 gia trị liên tiếp trong nhóm dữ liệu. Mặc định nó là 1 đơn vị, dù bạn không nhập vào Interval.
* Tham số đầu vào: Chỉ nhận đầu vào là Data và Interval.
* Kết quả trả về: nó trả về giá trị trung vị của tập hợp dữ liệu Data. Kết quả của chúng nằm trong khoảng Interval mà mình đã thiết lập.
* CODE:
* KẾT QUẢ:

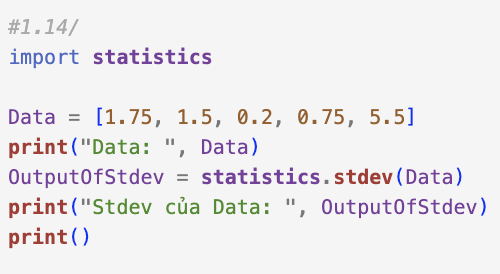
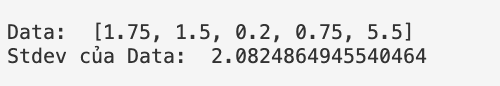
1. Hàm statistics.mode(Data)

* Hàm được sử dụng để tìm giá trị xuất hiện nhiều nhất trong một tập hợp dữ liệu trong Python.
* Data là một tập hợp các giá trị dùng để tìm mode. Data có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số, đôi khi cũng có các loại dữ liệu khác.
* Tham số đầu vào: Chỉ nhận duy nhất 1 tham số là Data.
* Kết quả trả về: Hàm trả về giá trị mà được xuất hiện đi xuất hiện lại nhiều nhất trong Data. Nếu có nhiều hơn một giá trị được cho là xuất hiện nhiều nhất, thì hàm sẽ trả về một trong các số đó. Đây là một hạn chế của nó, để khắc phục thì chúng ta có hàm multimode().
* CODE:****
* KẾT QUẢ:****

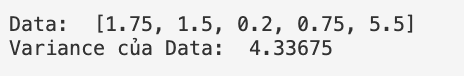
1.10 Hàm statistics.multimode(Data)

* Hàm được sử dụng để tìm tất cả các giá trị được cho là xuất hiện nhiều nhất trong một tập hợp dữ liệu Data đầu vào.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* Tham số đầu vào: Hàm chỉ nhận duy nhất 1 tham số vào là Data.
* Kết quả trả về: một danh sách chứa các giá trị mà xuất hiện nhiều nhất trong Data. Nếu có nhiều giá trị có số lần xuất hiện như nhau và nhiều nhất thì đều được trả về hết. Nó khắc phục được hạn chết của hàm mode() đã nêu ở trên.
* CODE:
* KẾT QUẢ:
  1. Hàm statistics.quantiles(Data, \*, n=4, method=’exclusive’)
* **Hàm** được sử dụng để tính các số lượng chia tách hoặc là phân vị của tập hợp dữ liệu Data. Các số lượng phân vị là các giá trị chia tập hợp thành các phần bằng nhau.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* n: tuỳ ý, nó là số lượng chia tách mà bạn muốn tinh ra. Mặc định n=4, nghĩa là hàm sẽ trả về các chia tách 25%, 50% và 75%.
* method: tuỳ chọn, đây là phương pháp tính do bạn muốn. Mặc định method=’exclusive’, nghĩa là phân vị được tính vào phương pháp loại trừ.
* Tham số đầu vào: Chỉ nhận 3 tham số là Data, n và method.
* Kết quả trả về: nó trả về danh sách gồm các phân vị, chia tách của tập Data của bạn.
* CODE:
* KẾT QUẢ:
  1. Hàm statistics.pstdev(Data, mu=None)
* Hàm thường được sử dụng khi tính độ lệch chuẩn dân số của toàn bộ tập hợp Data.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* mu: tuỳ chọn do bạn, nó là giá trị trung bình dân số của Data. Nếu nó mặc định là None hoặc không được cung cấp thì nó sẽ tính giá trị trung bình của dân số.
* Tham số đầu vào: Chỉ nhận Data và mu.
* Kết quả trả về: Hàm trả về độ lệch chuẩn tổng thể của tập hợp Data. Độ lệch chuẩn này nó thống kê mô tả mức độ phân tán của Data so với giá trị trung bình.
* CODE:
* KẾT QUẢ:
  1. Hàm statistics.pvariance(Data, mu=None)
* Hàm được sử dụng khi tính phương sai của tập hợp Data.
* Data: là một tập hợp các giá trị dùng để tính giá trị trung vị. Tham số có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số.
* mu: tuỳ chọn do bạn, nó là giá trị trung bình dân số của Data. Nếu nó mặc định là None hoặc không được cung cấp thì nó sẽ tính giá trị trung bình của dân số.
* Tham số đầu vào: Chỉ nhận đầu vào là mu và Data.
* Kết quả trả về: Hàm trả về phương sai dân số của Data. Nó là một đại lượng mà thống kê mức độ phân tán dữ liệu so với gía trị mu.
* CODE:
* KẾT QUẢ:

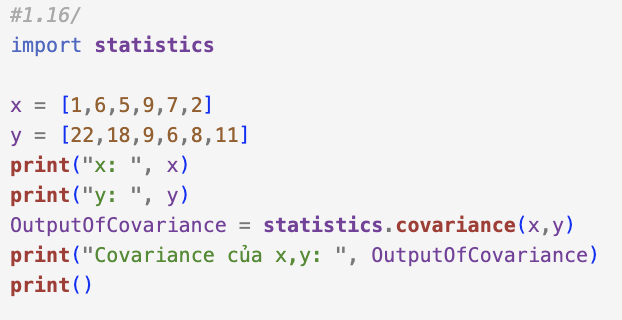
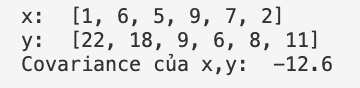
1.14 Hàm stdev()

* Hàm này trả về độ lệch chuẩn của dữ liệu, đây là một thước đo về mức độ biến đổi hoặc phân tán trong tập giá trị. Nó nhận một danh sách hoặc tuple của các số làm đầu vào.
* Data là một tập hợp các giá trị. Data có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số, đôi khi cũng có các loại dữ liệu khác.
* Tham số xbar là giá trị trung bình của tập hợp dữ liệu (tùy chọn). Nếu tham số này được cung cấp, hàm sẽ tính độ lệch chuẩn mẫu dựa trên giá trị trung bình đã biết trước xbar. Nếu xbar không được cung cấp (hoặc được đặt bằng None), hàm sẽ tự tính giá trị trung bình của data.
* Các tham số đầu vào: Nhận 2 tham số là Data và xbar
* Kết quả trả về: Hàm trả về độ lệch chuẩn mẫu của Data. Độ lệch chuẩn là một thước đo về mức độ phân tán của dữ liệu. Nó được tính bằng cách lấy căn bậc hai của phương sai.
* CODE:
* KẾT QUẢ:

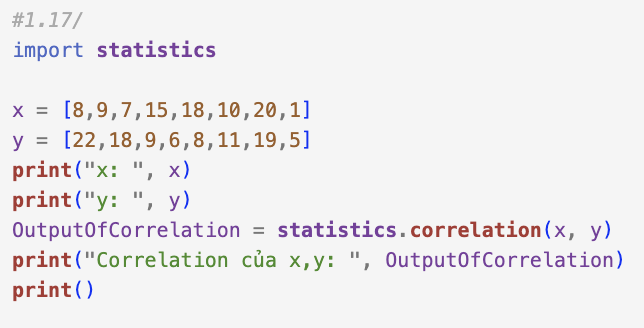
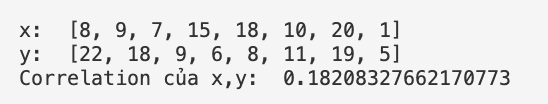
1.15 Hàm statistics.variance(data, xbar=None)

* Hàm này trả về phương sai của dữ liệu, đây là bình phương của độ lệch chuẩn.
* Data là một tập hợp các giá trị. Data có thể là một danh sách, một mảng, một tuple hoặc hoặc bất kỳ dạng iterable nào khác chứa các số, đôi khi cũng có các loại dữ liệu khác.
* xbar: tuỳ chọn, nó là giá trị trung bình của tập hợp dữ liệu . Nếu tham số này được cung cấp, hàm sẽ tính độ lệch chuẩn mẫu dựa trên giá trị trung bình đã biết trước xbar. Nếu xbar không được cung cấp hoặc được đặt bằng None, hàm sẽ tự tính giá trị trung bình của data.
* Các tham số đầu vào: Nhận 2 tham số là Data và xbar
* Kết quả trả về: Hàm này trả về phương sai của data. Phương sai là một thước đo về mức độ phân tán của dữ liệu.
* CODE:
* KẾT QUẢ:

1.16 Hàm statistics.covariance(x,y,/)

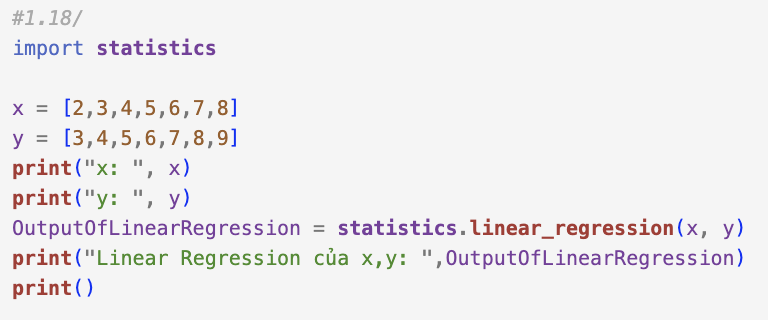
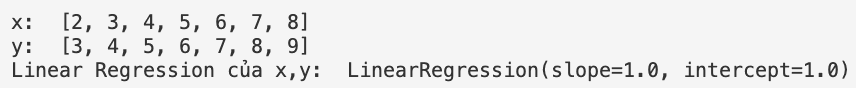
* Hàm thường được được sử dụng để tính độ lệch chung, hiệp phương sai giữa x và y do mình nhập vào.
* x: là tập hợp các giá trị của biến x, mà bạn muốn tính.
* y: là tập hợp các giá trị của biến y, mà bạn muốn tính.
* /: kí hiệu này ám chỉ rằng các tham số x,y không thể được chỉ định bằng tên.
* Tham số đầu vào: Chỉ nhận vào x và y là 2 tham số và có cùng độ dài.
* Kết quả trả về: : Hàm này trả về giá trị hiệp phương sai giữa 2 tham số do mình nhập là x và y.
* Hàm này mới xuất hiện từ phiên bản Python 3.10.
* CODE:
* KẾT QUẢ:

1.17 Hàm statistics.correlation(x,y,/)

* Hàm được sử dụng để tính hệ số tương quan giữa 2 tham số đầu vào là x và y. Hệ số tương quan thống kê mối quan hệ tuyến tính giữa 2 tham số đầu vào.
* Tham số đầu vào: Chỉ nhận duy nhất 2 tham số đầu vào là X và Y, có cùng độ dài.
* x: là tập hợp các giá trị của biến x, mà bạn muốn tính.
* y: là tập hợp các giá trị của biến y, mà bạn muốn tính.
* /: kí hiệu này ám chỉ rằng các tham số x,y không thể được chỉ định bằng tên.
* Kết quả trả về: nó trả về giá trị hệ số tương quan giữa x và y. Giá trị này nằm trong khoảng từ -1 đến +1, với +1 là tương quan dương hoàn hảo còn -1 là tương quan âm hoàn hảo. Còn số 0 là không có tương quan.
* Hàm này mới xuất hiện từ phiên bản Python 3.10.
* CODE:
* KẾT QUẢ:

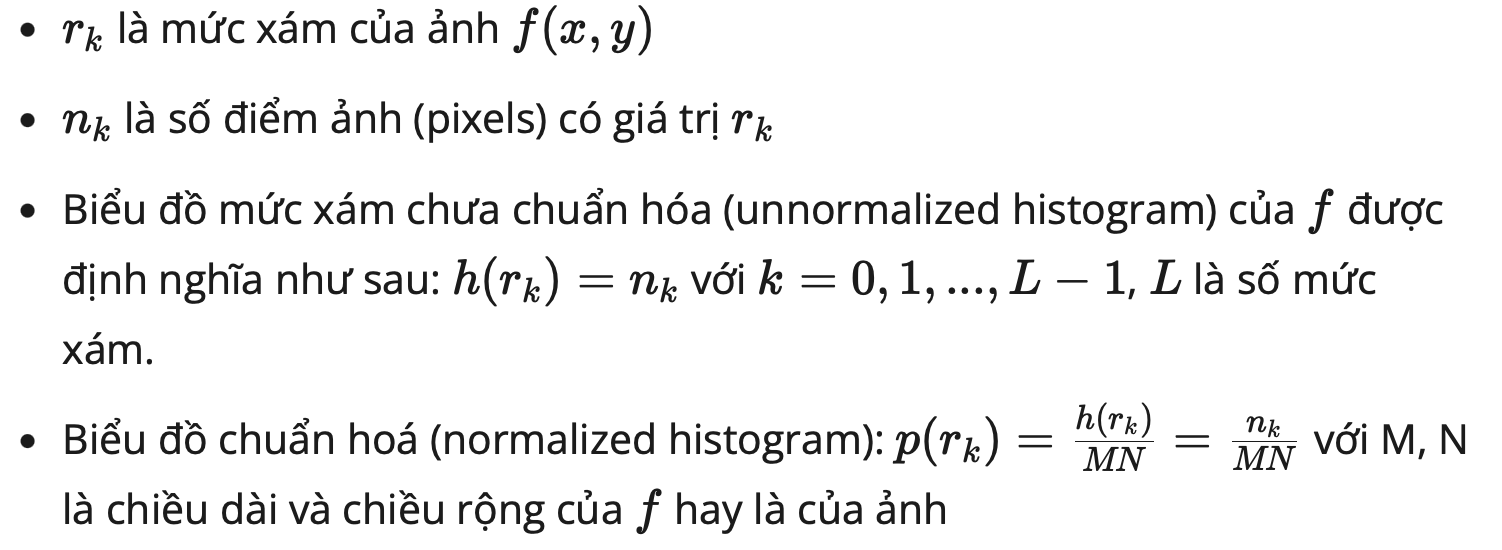
1.18 Hàm statistics.linear\_regression(x,y,/, \*,proportional=False)

* Hàm được sử dụng để tính toán mô hình hồi quy tuyến tính giữa x và y do mình nhập vào.
* x: là tập hợp các giá trị của biến x, mà bạn muốn tính.
* y: là tập hợp các giá trị của biến y, mà bạn muốn tính.
* Proportional: tuỳ chọn do mình muốn, nó trả về mô hình hồi quy tuyến tính thông qua góc độ độ O. Mặc định Proportional=False, nó sẽ trả về mô hình hồi quy mà không buộc phải thông qua góc toạ độ O.
* Tham số đầu vào: nhận 3 tham số đầu vào là x,y và proportional.
* Kết quả trả về: mô hình hồi quy tuyến tính giữa x và y. Nó mang hai thuộc tính là điểm cắt trục y và độ dốc của đường hồi quy.
* Hàm này mới xuất hiện từ phiên bản Python 3.10.

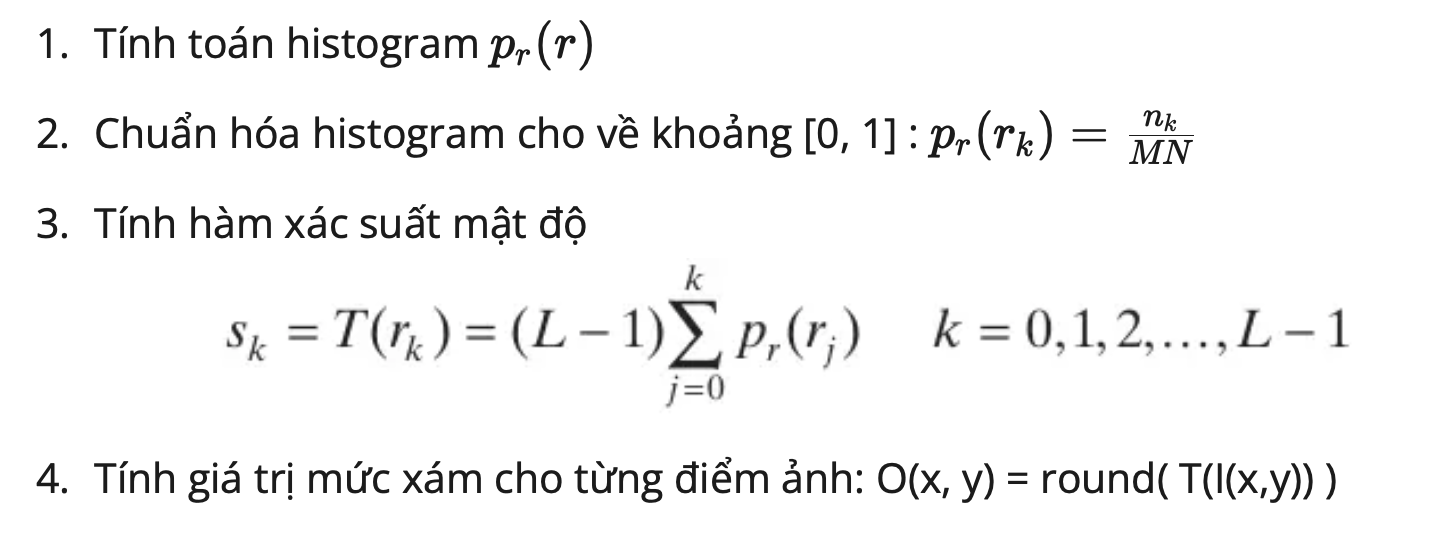
* CODE:
* KẾT QUẢ:

CHƯƠNG 2: THUẬT TOÁN HISTOGRAM EQUALIZATION

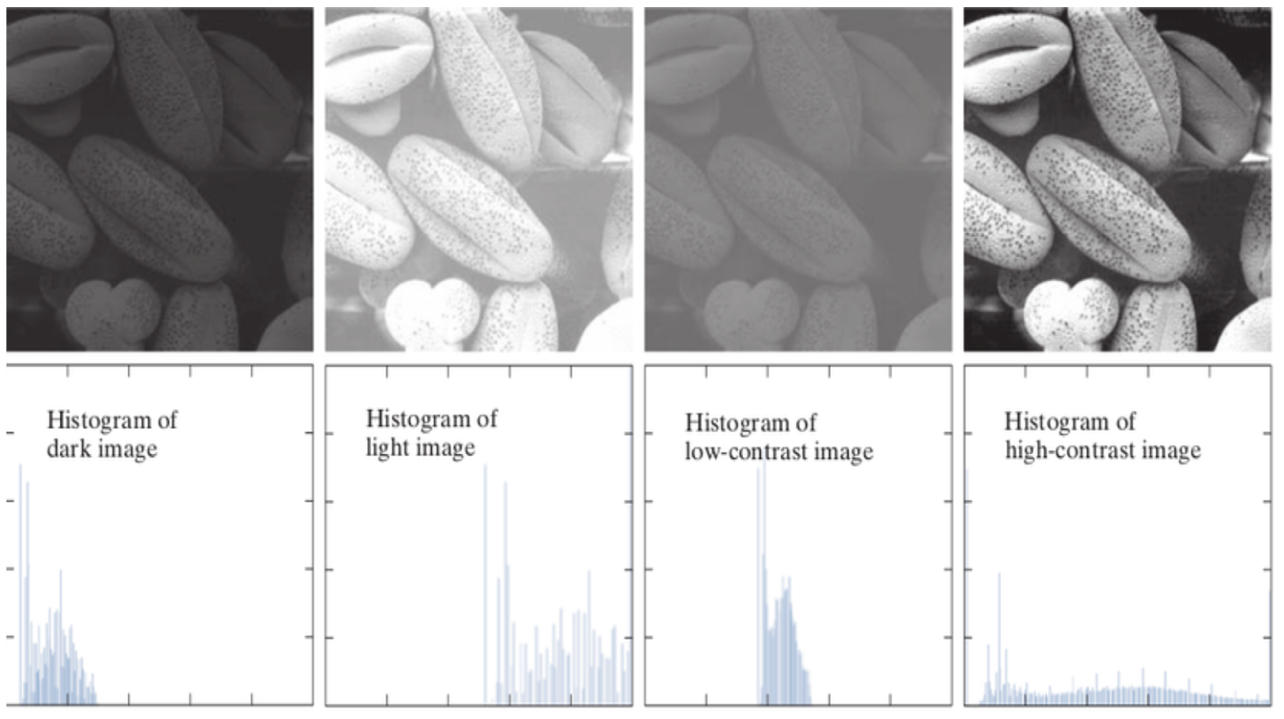
* 1. Khái niệm và vấn đề:
* Thuật toán Histogram equalization là thuật toán mà được áp dụng trong kĩ thuật xây dựng và xử lí hình ảnh, nó có tác dụng cải thiện độ , tăng độ tương phản của hình ảnh được cung cấp. Thuật toán thực hiện bằng cách hiệu quả lan rộng các giá trị cường độ phổ biến cao, nghĩa là nó sẽ kéo dãn phạm vi cường độ của hình ảnh. Phương pháp này rất hữu ích khi chúng ta xử lí các hình ảnh có nền và tiền cảnh đều sáng hoặc đều tối. Xử lý hình ảnh kỹ thuật số là một khía cạnh quan trọng của các bộ môn khoa học dữ liệu.
  1. Điều kiện và rằng buộc:
* Đảm bảo ảnh nhập vào là hình ảnh mức xám, nghĩa là ảnh chỉ có 1 kênh màu duy nhất. Trong hình ảnh mức xám ấy, mỗi pixel chỉ có 1 giá trị độ sáng, không có thông tin về màu sắc. Khi ấy sẽ giúp cho phép tính toán biểu đồ và áp dụng thuật toán trở nên hiệu quả và đơn giản hơn.
* Nếu ảnh là ảnh màu, thì chúng ta có thể thực hiện nó bằng cách chuyển ảnh sang một không gian màu khác, rồi áp dụng giải thuật cho cho kênh độ sáng, sau đó chuyển hình ảnh về lại không gian màu gốc.
* Giá trị mức xám nằm trong khoảng từ 0 đến 255, vì với hình ảnh mức xám 8 bits, mỗi pixel sẽ có giá trị từ 0 đến 255, với 0 là đen thui, còn 255 là trắng tinh. Thuật toán sẽ thực hiện với các giá trị pixel này để nâng cao chất lượng hình ảnh, cụ thể là cải thiện độ tương phản.
* Hạn chế của thuật toán là đôi khi nó có thể làm cho hình ảnh quá sáng.
* Thuật toán chỉ dựa vào biểu đồ của hình ảnh chứ không dựa vào nội dung cấu trúc của ảnh.
  1. Công thức tính Histogram toán học:



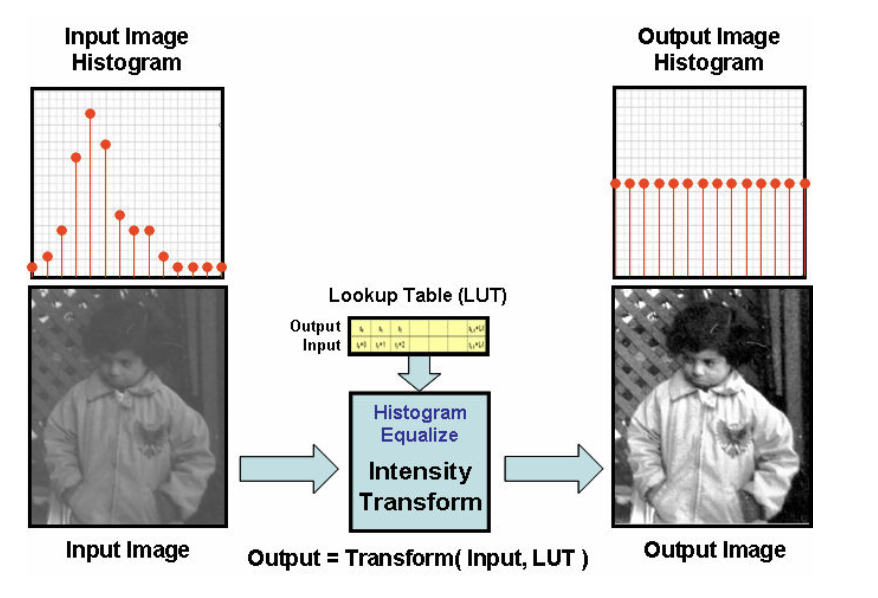
* 1. Công thức tính Histogram equalization toán học:



* 1. Thuật Toán:
* Đối với ảnh mức xám: Đầu tiên chúng ta phải đi tìm và tính toán biểu đồ của hình ảnh, biểu đồ nó sẽ thể hiện được tần số xuất hiện của các gía trị pixel. Tiếp đến, ta sẽ đi tìm hàm phân phối tích luỹ của biểu đồ hình ảnh, hàm phân phối tích luỹ sẽ thể hiện được tổng tần số của các giá trị pixel có được mà nhỏ hơn hoặc bằng giá trị pixel hiện tại. Bước cuối cùng là sử dụng hàm phân phối tích luỹ ta vừa có được để ánh xạ các giá trị pixel gốc của ảnh thành các giá trị mớ, làm cho biểu đồ ảnh được cân bằng, cải thiện hơn.
* Đối với ảnh màu: Đầu tiên chúng ta phải biến đổi ảnh từ 1 không gian xanh dương, xanh lá và đỏ sang không gian màu YcrCb (Y là độ sáng, Cr là đại diện cho sự khác biệt giữa màu đỏ và độ sáng Y, Cb thì lại đạ diện cho sự khác biệt giữa màu xanh dương và độ sáng Y). Tiếp đến ta sẽ tiến hành tách hình ảnh làm 3 là Y, Cr,Cb. Sau đó mới áp dụng thuật toán cân bằng giống như ở cân bằng mức xám. Sau khi áp dụng xong thuật toán, ta sẽ hợp nhất chúng gồm Y,Cr,Cb lại với nhau từ đó chuyển đổi được hình ảnh quay trở lại không gian màu xanh dương, xanh lá và đỏ như ban đầu.
  1. Ví dụ:
* Histogram:



* Histogram Equalization:



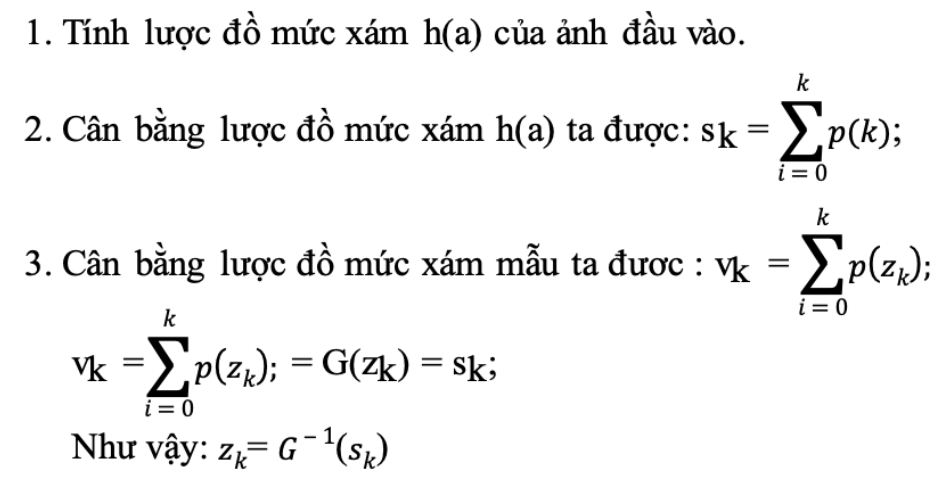


Ảnh gốc Ảnh khi áp dụng thuật toán

* 1. Nhận xét:
* Thuật toán Histogram Equalization là 1 phương pháp rất phổ biến, hiệu quả và đơn giản trong việc nâng cao chất lượng tương phản của hình ảnh. Nó có thể áp dụng cho cả hình ảnh 1 kênh màu hay nhiều kênh màu khác nhau. Thuật toán không yêu cầu nguồn vào như thuật toán Histogram matching, nhờ đó nó lại tiện lợi dễ dùng hơn. Thuật toán rất phù hợp cho các ảnh tối hoặc quá sáng. Tuy rất hiệu quả nhưng nó còn có hạn chế chẳng hạn như nó có thể làm cho hình ảnh trở nên quá sáng, làm mất đi nhiều điểm trong hình ảnh. Nếu ảnh ban đầu mang các yếu tố nhiễu thì nó có thể sẽ làm tăng thêm nhiễu, làm cho chất lượng ảnh ngày càng tệ đi. Hình ảnh sau khi áp dụng thuật toán không phải lúc nào cũng cho ra kết quả tốt.

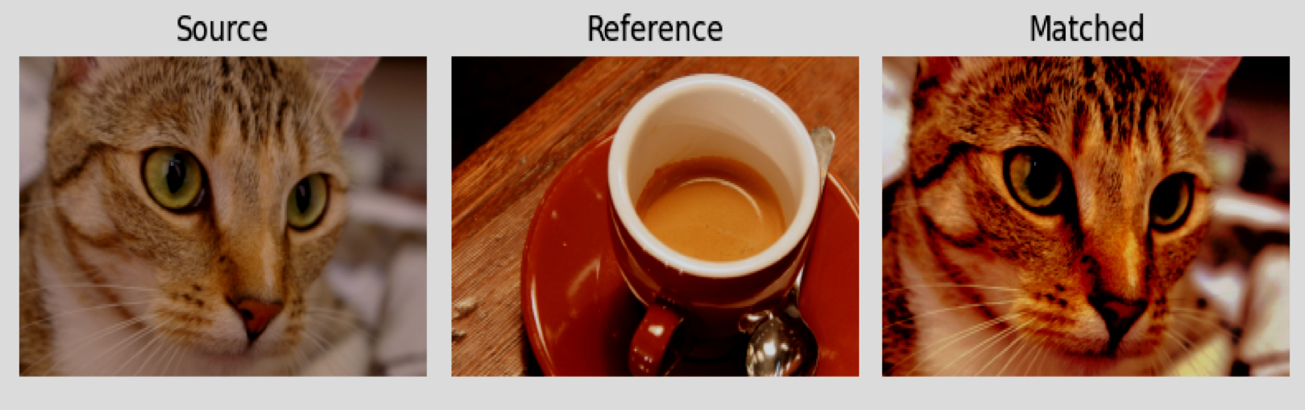
CHƯƠNG 3: THUẬT TOÁN HISTOGRAM MATCHING

* 1. Khái niệm và vấn đề:
* Histogram Matching là một phương pháp sử dụng để chuẩn hóa biểu đồ mức xám của hình ảnh, đặc biệt khi ảnh đến từ các nguồn khác nhau hoặc trong điều kiện ánh sáng và môi trường khác nhau. Phương pháp là 1 quá trình điều chỉnh biểu đồ của ảnh đầu vào sao cho trùng khớp với biểu đồ ảnh của ảnh mục tiêu, nói cách khác là cân bằng lại mức xám sao cho ảnh đầu vào phù hợp với mức xám của ảnh mục tiêu. Giải thuật này có thể được áp dụng cho các kênh màu của ảnh.
  1. Rằng buộc và điều kiện:
* Số lượng kênh của ảnh đầu vào và ảnh mục tiêu phải giống nhau.
* Kích thước của các ảnh phải bằng nhau, nghĩa là có cùng kích thước như nhau.
* Thuật toán không thể xử lý được sự thay đổi độ sáng, nó chỉ điều chỉnh phân phối màu sắc mà thôi.
* Thuật toán cũng không thể xử lý được sự thay đổi độ tương phản.
* Kết quả của thuật toán không được ổn định, nó phụ thuộc vào ảnh mục tiêu.
  1. Công thức toán học:



* 1. Thuật toán:
* Đầu tiên ta sẽ đi tìm và tính toán biểu đồ ảnh của hình ảnh đầu vào, sau đó là đi tính biểu đồ tích luỹ của chúng.
* Sau đó chúng ta sẽ tạo ra 1 bảng được gọi là bảng tra cứu LUT, hay được gọi là hàm M(). Trong đó ánh xạ các giá trị màu sắc tương ứng của ảnh đầu và ảnh mục tiêu.
* Ta sử dụng bảng LUT hay gọi là áp dụng hàm Matching M() lên ảnh gốc để thay thế các giá trị màu sắc tương ứng của ảnh đầu vào và ảnh mục tiêu. Kết quả sẽ là hình ảnh đầu có biểu đồ ảnh giống với ảnh mục tiêu.
  1. Ví dụ:

Ví dụ 1



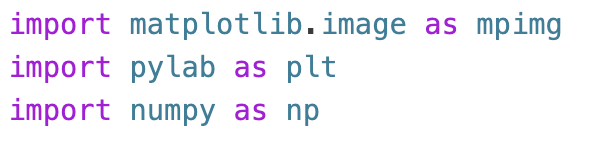
Ví dụ 2



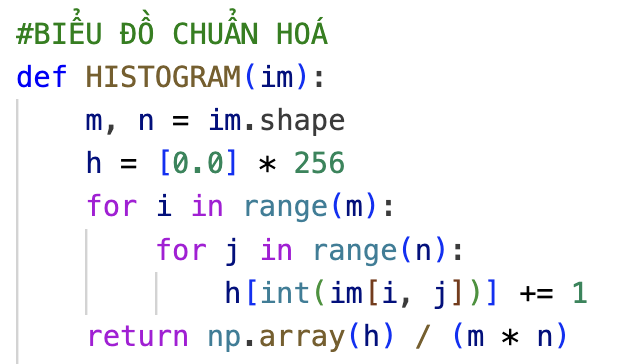
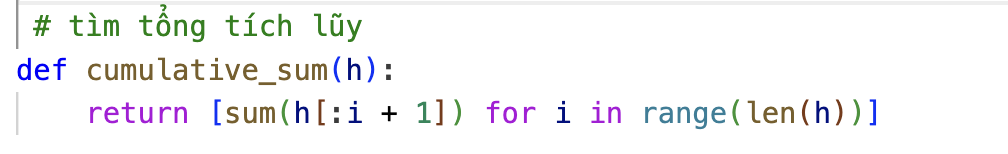
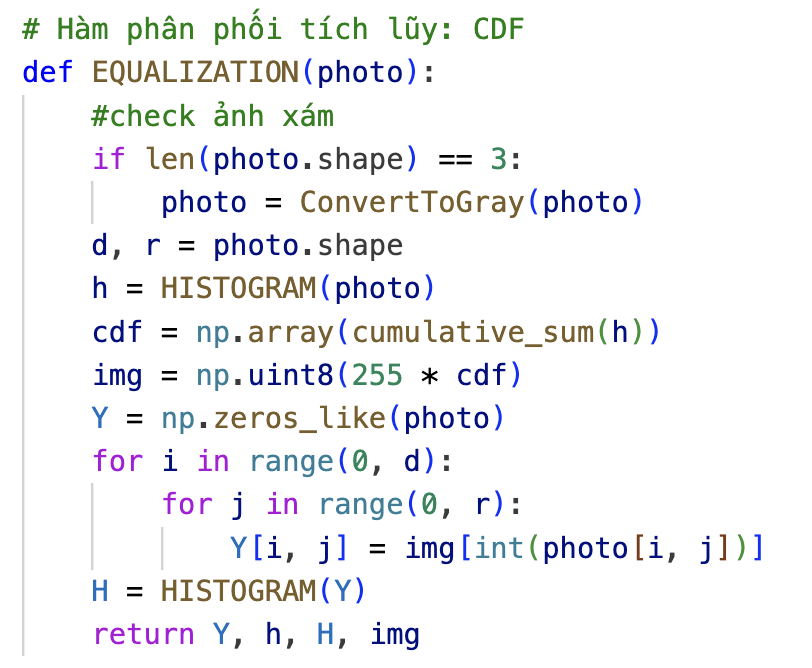
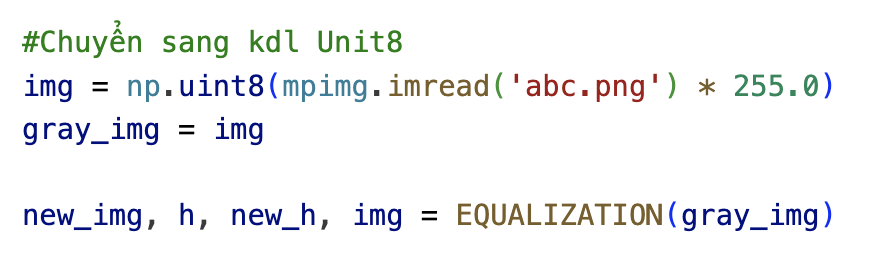
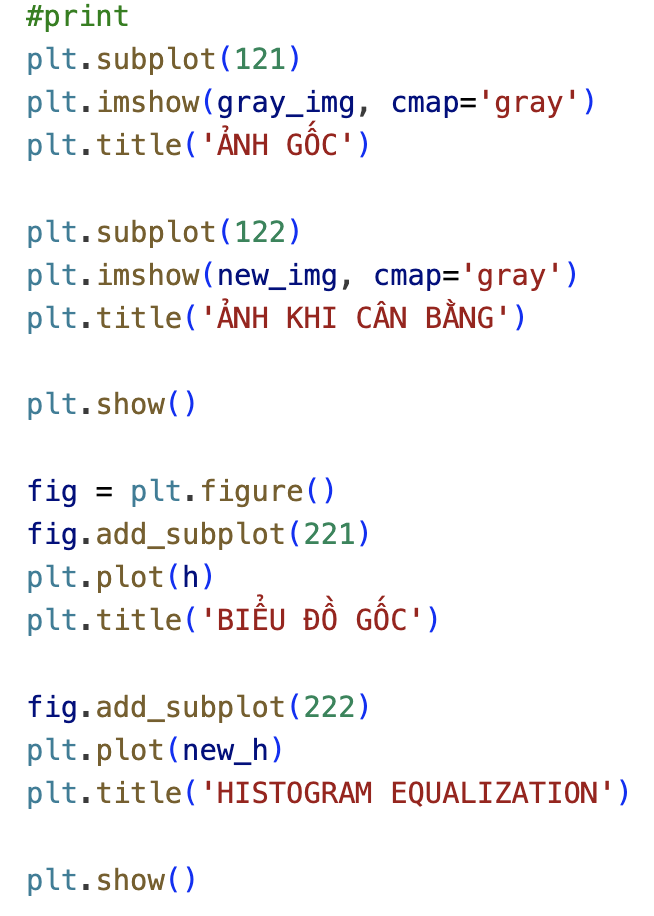
* 1. Nhận xét:
* Thuật toán Histogram Matching là 1 giải thuật rất mạnh mẽ và đặc biệt quan trọng trong việc chuẩn hoá, xử lí và nâng cao chất lượng hình ảnh. Nó có sẵn các thư viện trong python và trong nhiều ngôn ngữ khác nhờ đó nó lại trở nên dễ sử dụng. Dù mang nhiều lợi ích, nhưng nó lại có rất nhiều mặt hạn chế, Thuật toán không thể xử lý được sự thay đổi độ sáng, độ tương phản. Kết quả của thuật toán không được ổn định, nó phụ thuộc vào ảnh mục tiêu, mục tiêu mà bất ổn dễ dẫn đến hỏng hết ảnh ban đầu. Nếu ảnh ban đầu mang các yếu tố nhiễu thì nó có thể sẽ làm tăng thêm nhiễu, làm cho chất lượng ảnh ngày càng tệ đi.

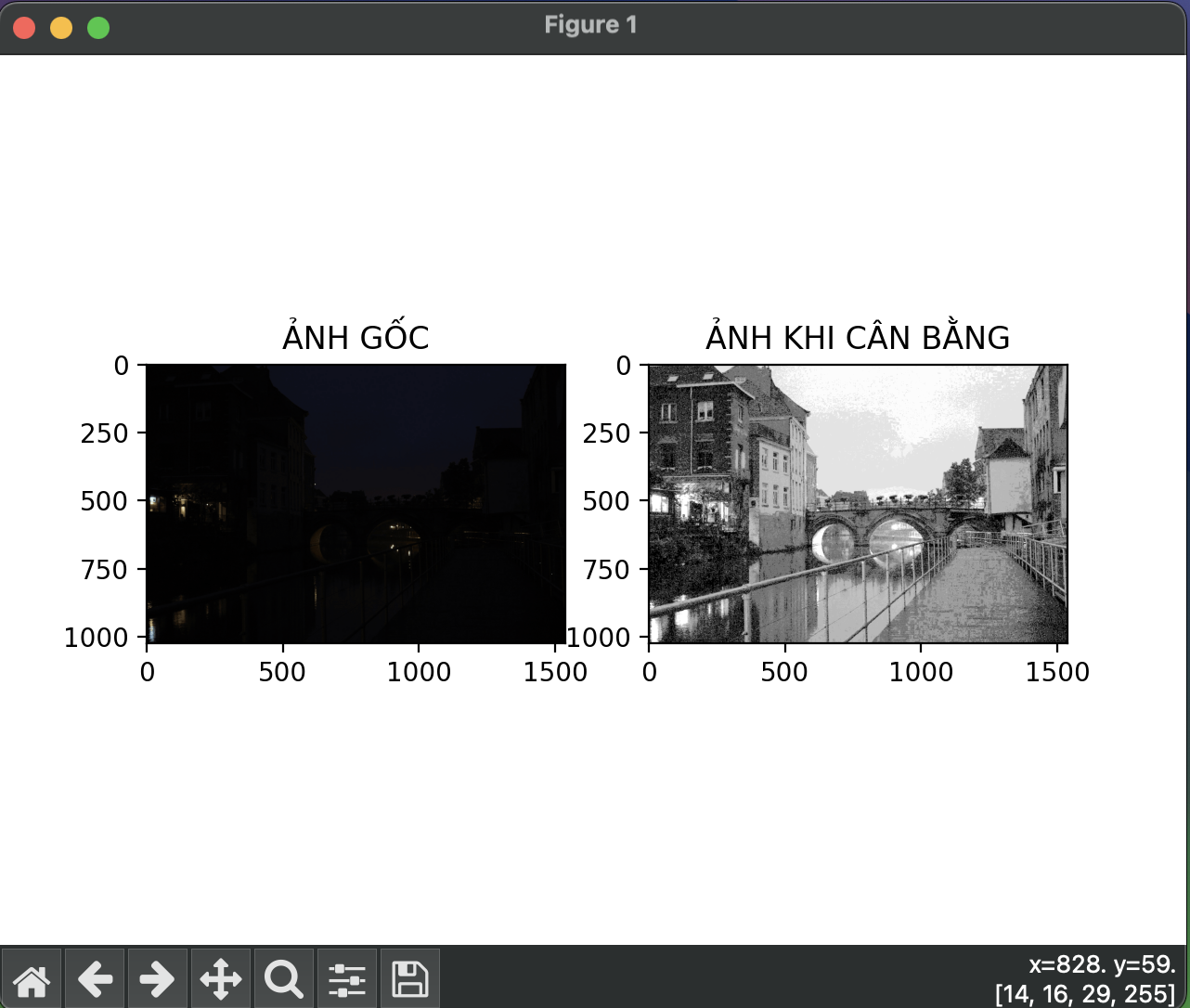
CHƯƠNG 4: THỰC HIỆN GIẢI THUẬT

1. HISTOGRAM EQUALIZATION:

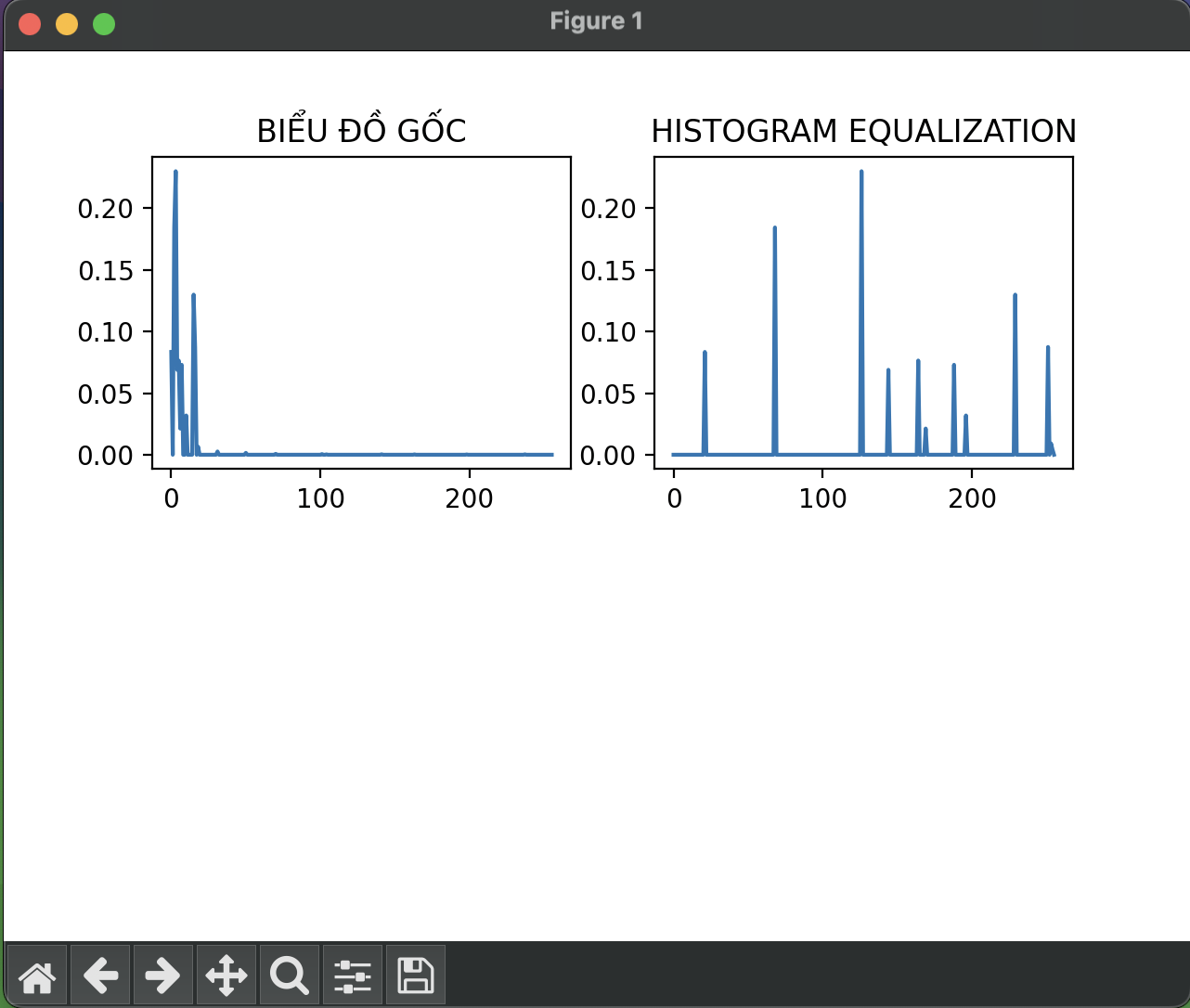
* Cần dùng 3 thư viện là Matplotlib, Numpy và Pylab. 
* Cần 1 hàm chuyển đổi ảnh màu sang ảnh xám



* Cần 1 hàm biểu đồ histogram: 
* Cần 1 hàm tính tổng tích luỹ:
* Cần 1 hàm CDF:
* Tiến hành nhập ảnh vào và áp dụng thuật toán (chỉ ảnh png):
* Sau khi cân bằng, in ảnh và biểu đồ ra bằng thư viện Matplotlib:
* Bước cuối cùng chỉ việc chạy file .py, chúng ta sẽ có các kết quả sau, với hình bên trái là ảnh gốc, bên phải là ảnh sau khi chuyển:

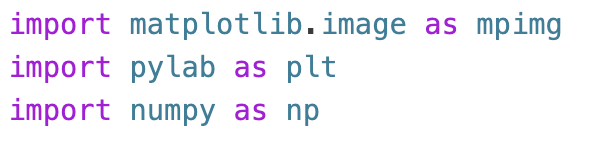


*So sánh 2 bức ảnh*

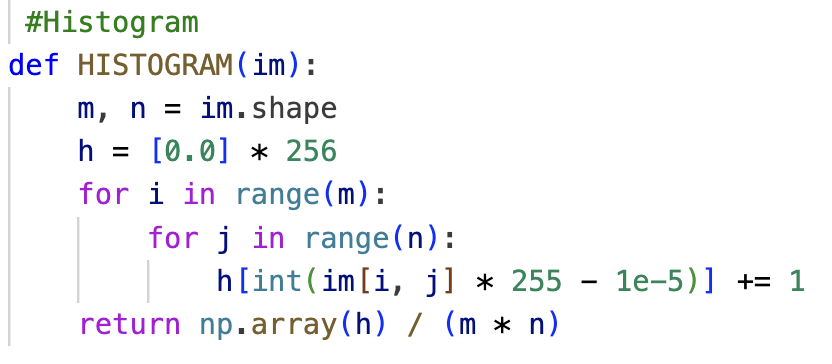
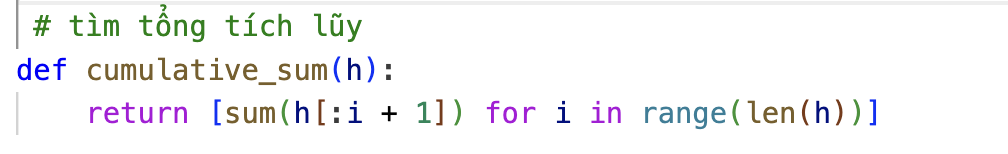
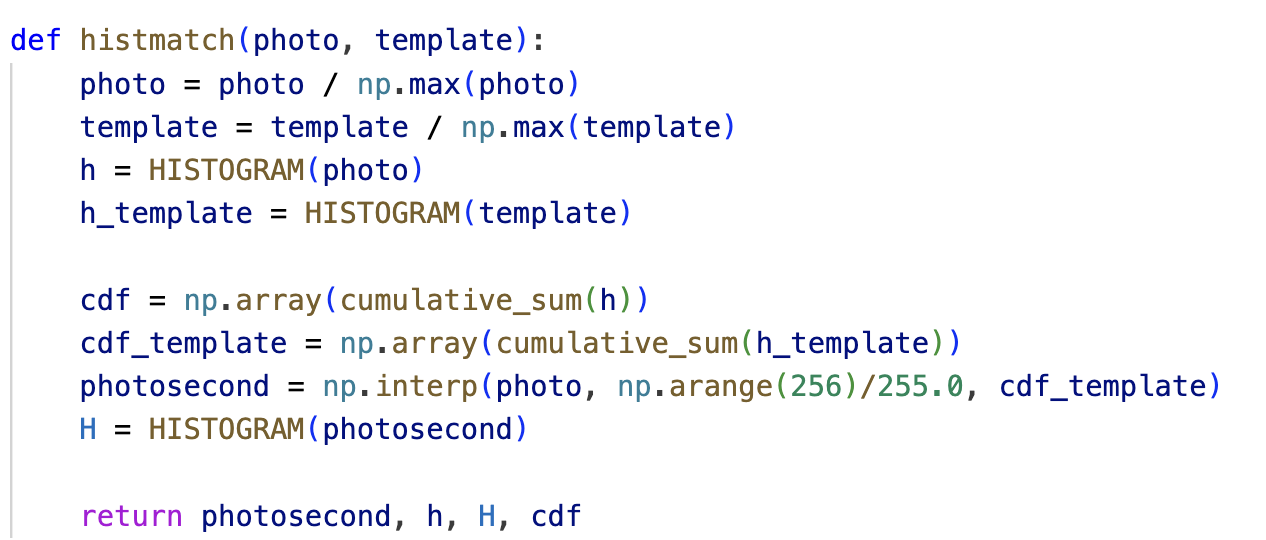
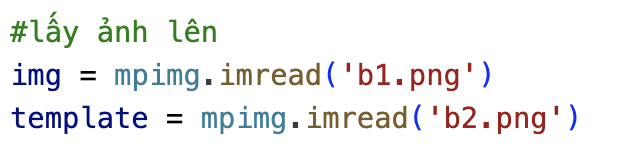


So sánh 2 biểu đồ để thấy sự tương quan

1. HISTOGRAM MATCHING:

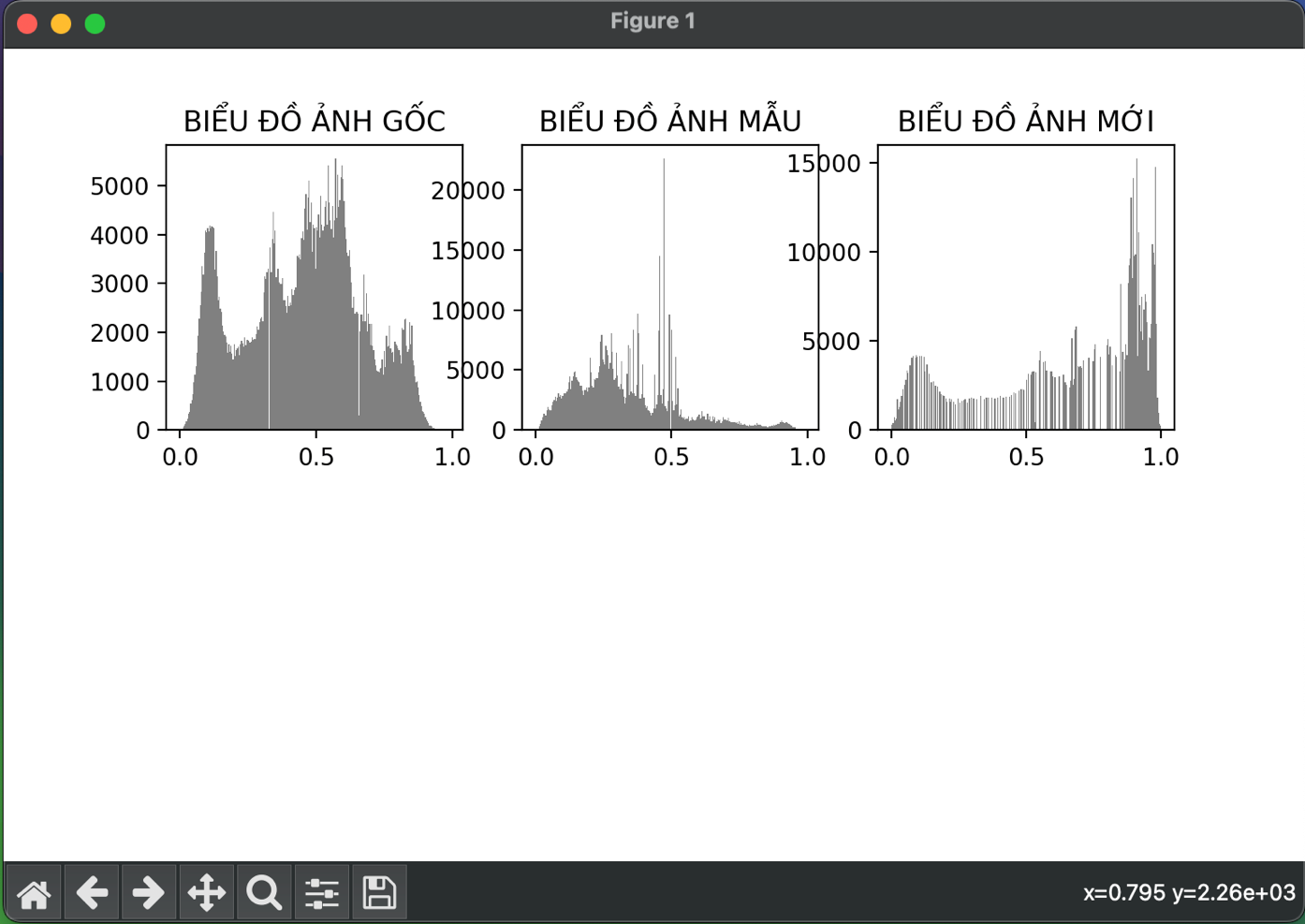
* Cần dùng 3 thư viện là Matplotlib, Numpy và Pylab. 
* Cần 1 hàm chuyển đổi ảnh màu sang ảnh xám



* Cần 1 hàm biểu đồ histogram: 
* Cần 1 hàm tính tổng tích luỹ:
* Cần 1 hàm HISTOGRAM MATCHING:
* Tiến hành nhập 2 ảnh vào (chỉ ảnh png): 
* Sau khi cân bằng, in ảnh và biểu đồ ra bằng thư viện Matplotlib:
* Bước cuối cùng chỉ việc chạy file .py, chúng ta sẽ có các kết quả sau, với hình bên trái là ảnh gốc, ở giữa là ảnh mẫu và bên phải là ảnh sau khi chuyển:



*So sánh các ảnh*



*So sánh các biểu đồ histogram*

TÀI LIỆU THAM KHẢO

**Các hàm được lấy thông tin và trích xuất từ:**

1. <https://docs.python.org/3/library/statistics.html>
2. <https://vietnambiz.vn/trung-vi-median-la-gi-vi-du-ve-trung-vi-2019110713491368.htm>
3. <https://vietnambiz.vn/trung-binh-nhan-geometric-mean-la-gi-vi-du-ve-trung-binh-nhan-20191114154522837.htm>
4. <https://minhng.info/tutorials/xu-ly-anh-opencv-can-bang-sang-histogram-equalization.html>
5. <https://viblo.asia/p/tuan-3-histogram-histogram-equalization-3P0lPnxmKox>
6. https://www.educative.io/answers/what-is-histogram-equalization-in-python
7. <https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html>
8. <https://en.wikipedia.org/wiki/Histogram_equalization>
9. <https://en.wikipedia.org/wiki/Histogram_matching>
10. <https://namlunoy.wordpress.com/2015/01/26/histogram-matching/>
11. https://i.stack.imgur.com/GWtt1.jpg