**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÀI TIỂU LUẬN MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**TÌM HIỂU WEB SECURITY**

*Người hướng dẫn*: **Thầy VŨ ĐÌNH HỒNG**

*Người thực hiện*: **PHẠM ĐỨC MINH HIẾU – 52100796**

**NGUYỄN VĂN HÀO – 52100792**

Lớp **: 21050201**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÀI TIỂU LUẬN MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**TÌM HIỂU WEB SECURITY**

*(Bìa phụ)*

*Người hướng dẫn*: **Thầy VŨ ĐÌNH HỒNG**

*Người thực hiện*: **PHẠM ĐỨC MINH HIẾU – 52100796**

**NGUYỄN VĂN HÀO – 52100792**

Lớp **: 21050201**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Lời đầu tiên em xin được phép cảm ơn quý thầy cô khoa Công nghệ Thông Tin trường Đại học Tôn Đức Thắng đã tạo mọi điều kiện cho em tạo mọi điều kiện tốt nhất để em được học tập môn Xác suất và thống kê ứng dụng cho công nghệ thông tin.

Em cũng xin được gửi lời cảm ơn đến thầy Nguyễn Lâm, thầy đã nhiệt tình giảng dạy, trang bị đầy đủ kiến thức để em có thể hoàn thành bài tiểu luận giữa kỳ này.

Cuối cùng, do hạn chế về mặt kiến thức, kính mong thầy cô có thể bỏ qua những sai sót nhỏ và chỉ ra được những lỗi sai của em trong bài báo cáo này để những bài báo cáo sau của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn thầy Nguyễn Lâm và toàn thể quý thầy cô khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng.

# 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)

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc118666639)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 2](#_Toc118666640)

DANH MỤC HÌNH ẢNH VÀ BẢNG…………………………………………………5

[CHƯƠNG 1 – THƯ VIỆN STATISTICS 6](#_Toc118666641)

[1.1. Tổng quan về thư viện statistics trong Python: 6](#_Toc118666642)

[1.2. Hàm mean(): 6](#_Toc118666643)

[1.3. Hàm fmean(): 6](#_Toc118666644)

[1.4. Hàm geometric mean(): 7](#_Toc118666645)

[1.5. Hàm harmonic\_mean(): 8](#_Toc118666646)

[1.6. Hàm median(): 8](#_Toc118666647)

[1.7. Hàm median\_low(): 9](#_Toc118666648)

[1.8. Hàm median\_high(): 10](#_Toc118666649)

[1.9. Hàm median\_grouped(): 11](#_Toc118666650)

[1.10. Hàm mode(): 12](#_Toc118666651)

[1.11. Hàm multimode(): 13](#_Toc118666652)

[1.12. Hàm quantiles(): 13](#_Toc118666653)

[1.13. Hàm pstdev(): 14](#_Toc118666654)

[1.14. Hàm pvariance(): 15](#_Toc118666655)

[1.15. Hàm stdev(): 15](#_Toc118666656)

[1.16. Hàm variance(): 16](#_Toc118666657)

[1.17. Hàm covariance(): 17](#_Toc118666658)

[1.18. Hàm correlation(): 18](#_Toc118666659)

[1.19. Hàm linear\_regression(): 19](#_Toc118666660)

[CHƯƠNG 2 – THUẬT TOÁN HISTOGRAM ĐỂ XỬ LÝ ẢNH 20](#_Toc118666661)

[2.1. Vấn đề: 20](#_Toc118666662)

[2.2. Những ràng buộc và điều kiện của thuật toán: 20](#_Toc118666663)

[2.3. Thuật toán cân bằng Histogram 21](#_Toc118666664)

[2.4. Ví dụ 22](#_Toc118666665)

[2.5. Nhận xét, đánh giá 23](#_Toc118666666)

[CHƯƠNG 3: THỰC HIỆN 24](#_Toc118666667)

[3.1. Thư viện statistics trong Python 24](#_Toc118666668)

[3.2. Thuật toán cân bằng Histogram 24](#_Toc118666669)

[DANH MỤC TÀI LIỆU THAM KHẢO VÀ CÁC TRÍCH DẪN 27](#_Toc118666670)

**DANH MỤC HÌNH ẢNH VÀ BẢNG**

[*Hình 1. Sourcecode về ví dụ cho hàm mean* 6](#_Toc118665752)

[*Hình 2. Sourcecode về ví dụ cho hàm fmean* 7](#_Toc118665753)

[*Hình 3. Sourcecode về ví dụ cho hàm geometric\_mean* 8](#_Toc118665754)

[*Hình 4. Sourcecode về ví dụ cho hàm harmonic\_mean* 8](#_Toc118665755)

[*Hình 5. Sourcecode về ví dụ cho hàm median* 9](#_Toc118665756)

[*Hình 6. Sourcecode về ví dụ cho hàm median\_low* 10](#_Toc118665757)

[*Hình 7. Sourcecode về ví dụ cho hàm median\_high* 11](#_Toc118665758)

[*Hình 8. Sourcecode về ví dụ cho hàm median\_grouped* 12](#_Toc118665759)

[*Hình 9. Sourcecode về ví dụ cho hàm mode* 12](#_Toc118665760)

[*Hình 10. Sourcecode về ví dụ cho hàm multimode* 13](#_Toc118665761)

[*Hình 11. Sourcecode về ví dụ cho hàm quantiles* 14](#_Toc118665762)

[*Hình 12. Sourcecode về ví dụ cho hàm pstdev* 14](#_Toc118665763)

[*Hình 13. Sourcecode về ví dụ cho hàm pvariance* 15](#_Toc118665764)

[*Hình 14. Sourcecode về ví dụ cho hàm stdev* 16](#_Toc118665765)

[*Hình 15. Sourcecode về ví dụ cho hàm variance* 17](#_Toc118665766)

[*Hình 16. Sourcecode về ví dụ cho hàm covariance* 18](#_Toc118665767)

[*Hình 17. Sourcecode về ví dụ cho hàm correlation* 19](#_Toc118665768)

[*Hình 18. Sourcecode về ví dụ cho hàm linear\_regression* 20](#_Toc118665769)

[*Hình 19. Ma trận của bức ảnh có mức sáng từ 0 đến 7 pixel* 22](#_Toc118665770)

[*Bảng 1. Chuẩn hóa các giá trị pixel, tăng độ tương phản.* 23](#_Toc118665771)

[*Hình 20. Ma trận của bức ảnh sau khi trải qua thuật toán cân bằng Histogram* 23](#_Toc118665772)

[*Hình 21. Thuật toán cân bằng Histogram khi gặp nhiều dữ liệu nhiễu* 24](#_Toc118665773)

[*Hình 22. Ảnh trước khi xử lí* 2](#_Toc118665773)5

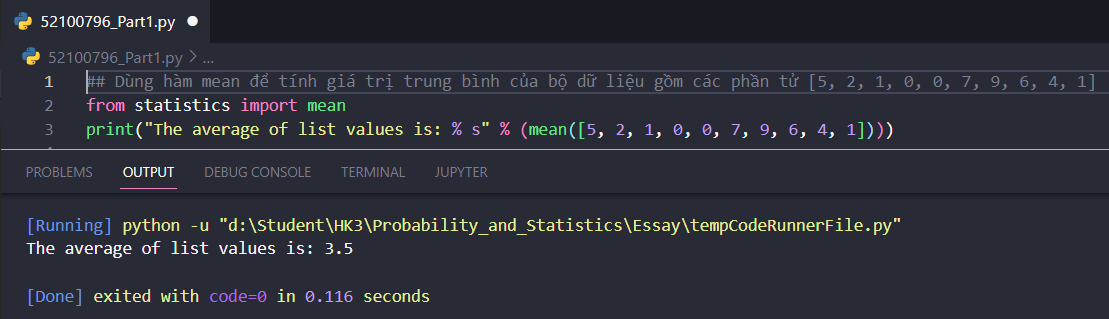
[*Hình 23. Lệnh đọc ảnh và hàm tính toán cho việc xử lí ảnh* 26](#_Toc118665774)

[*Hình 24. Hình ảnh được xử lí bằng thuật toán cân bằng histogram* 27](#_Toc118665775)

1. NHỮNG LỖ HỔNG THƯỜNG GẶP ĐỐI VỚI MỘT WEBSITE
   1. Tổng quan về thư viện statistics trong Python:

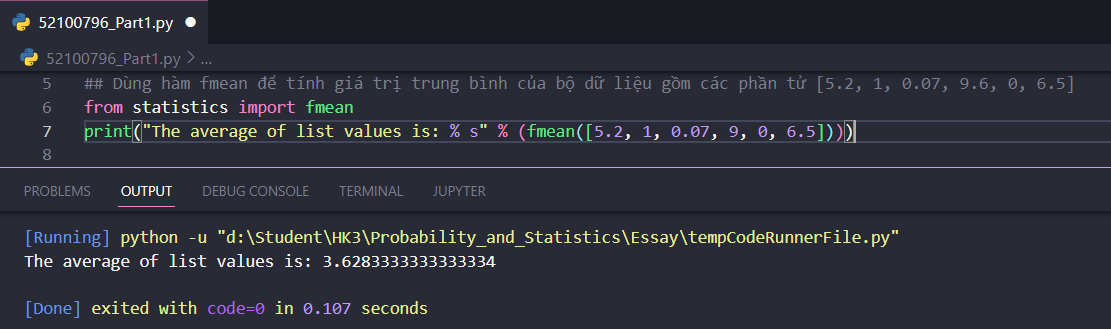
Thống kê ngày nay đã là một phần quan trọng trong đời sống và công việc của chúng ta. Và quan trọng hơn, với sự khổng lồ và đa dạng của dữ liệu hiện tại. Các công cụ tính toán thông thường dường như đã không đủ để đáp ứng các nhu cầu công việc hàng ngày. Đặc biệt thống kê đã được ứng dụng rất nhiều trong lĩnh vực Công nghệ thông tin. Vì thế, ngôn ngữ lập trình Python đã trình soạn thư viện statistics với mong muốn cải thiện hiệu suất công việc cũng như số liệu đưa ra sẽ được chính xác hơn. Dưới đây là một số hàm tính toán của thư viện statistics trong Python.

* 1. Hàm [mean()](https://docs.python.org/3/library/statistics.html#statistics.mean):
* *Công dụng:* hàm mean(data) để tính giá trị trung bình của một bộ dữ liệu.
* *Tham số đầu vào:* là một danh sách dữ liệu, thường mang kiểu list.
* *Ý nghĩa của kết quả:* kết quả trả về là tổng của tất cả phần tử trong danh sách dữ liệu chia cho số phần tử của danh sách dữ liệu đó.
* *Ví dụ:*



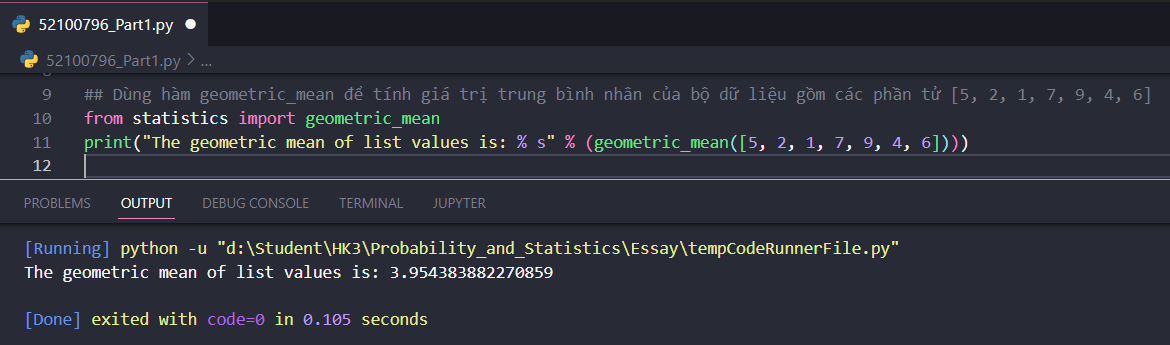
*Hình 1. Sourcecode về ví dụ cho hàm mean*

* 1. **Hàm fmean():**
* *Công dụng:* hàm fmean(data) dùng để tính giá trị trung bình của bộ dữ liệu, nhưng kết quả trả về luôn ở kiểu dữ liệu float. Hàm fmean thực thi nhanh hơn hàm mean.
* *Tham số đầu vào:* là một danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* kết quả trả về là tổng của tất cả phần tử trong danh sách dữ liệu chia cho số phần tử của danh sách dữ liệu đó, kết quả này mang kiểu dữ liệu float.
* *Ví dụ:*

**

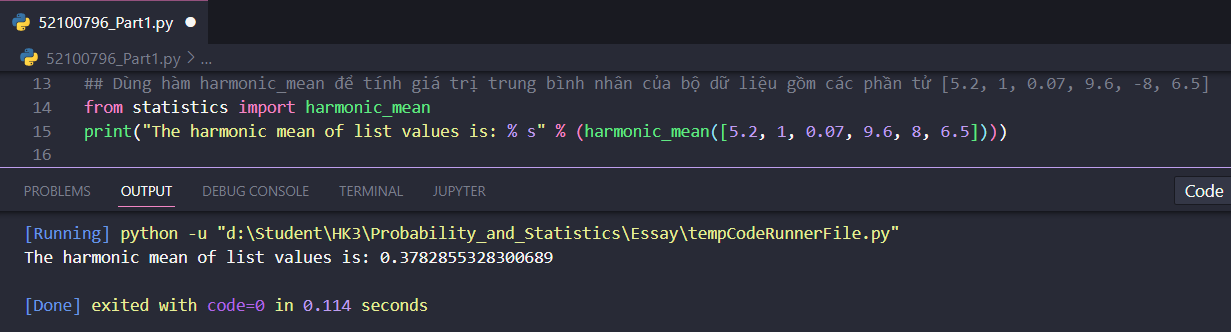
*Hình 2. Sourcecode về ví dụ cho hàm fmean*

* 1. **Hàm** [**geometric mean()**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm geometric\_mean(data) dùng để tính giá trị trung bình hình học của danh sách dữ liệu, nó còn có tên gọi khác là trung bình nhân.
* *Tham số đầu vào:* là một danh sách dữ liệu kiểu list, yêu cầu từng phần tử trong danh sách phải lớn hơn 0.
* *Ý nghĩa của kết quả:* kết quả trả về căn bậc n của tích n các giá trị nằm trong danh sách dữ liệu.
* *Ví dụ:*

**

*Hình 3. Sourcecode về ví dụ cho hàm geometric\_mean*

* 1. **Hàm** [**harmonic\_mean()**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm harmonic\_mean(data) dùng để tính trung bình điều hòa của một danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list, yêu cầu từng phần tử trong danh sách phải lớn hơn 0.
* *Ý nghĩa của kết quả:* kết quả trả về là nghịch đảo của tổng nghịch đảo của các phần tử trong danh sách dữ liệu.
* *Ví dụ:*

**

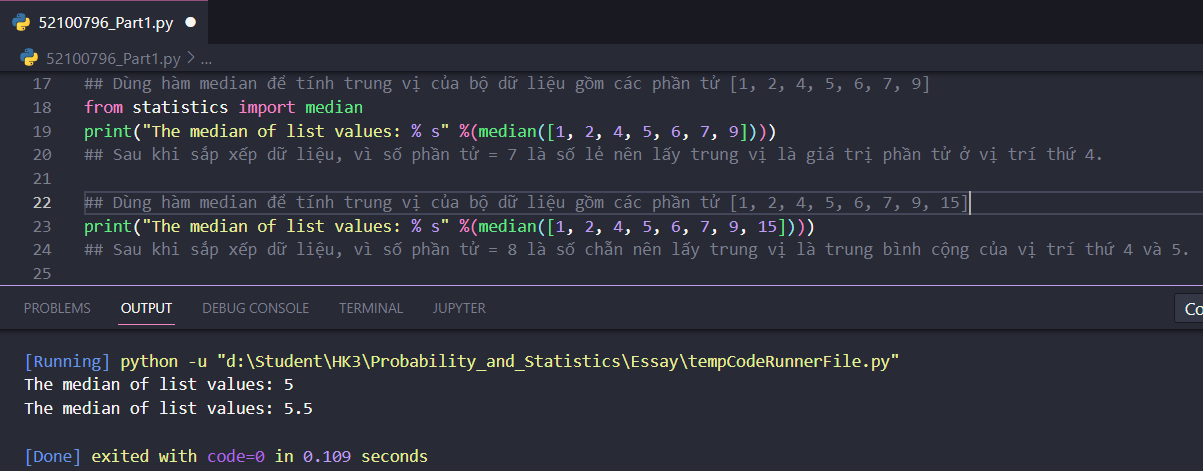
*Hình 4. Sourcecode về ví dụ cho hàm harmonic\_mean*

* 1. **Hàm**[[**median()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm median(data) dùng để tính trung vị của bộ dữ liệu. Trung vị là giá trị của phần tử nằm ở giữa danh sách sau khi đã sắp xếp dữ liệu tăng dần.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* kết quả trả về là giá trị chính giữa sau khi đã sắp xếp dữ liệu. Có hai trường hợp:

*+* Nếu số phần tử của danh sách dữ liệu là số lẻ, hàm sẽ trả về kết quả là vị trí thứ [n/2]+1.

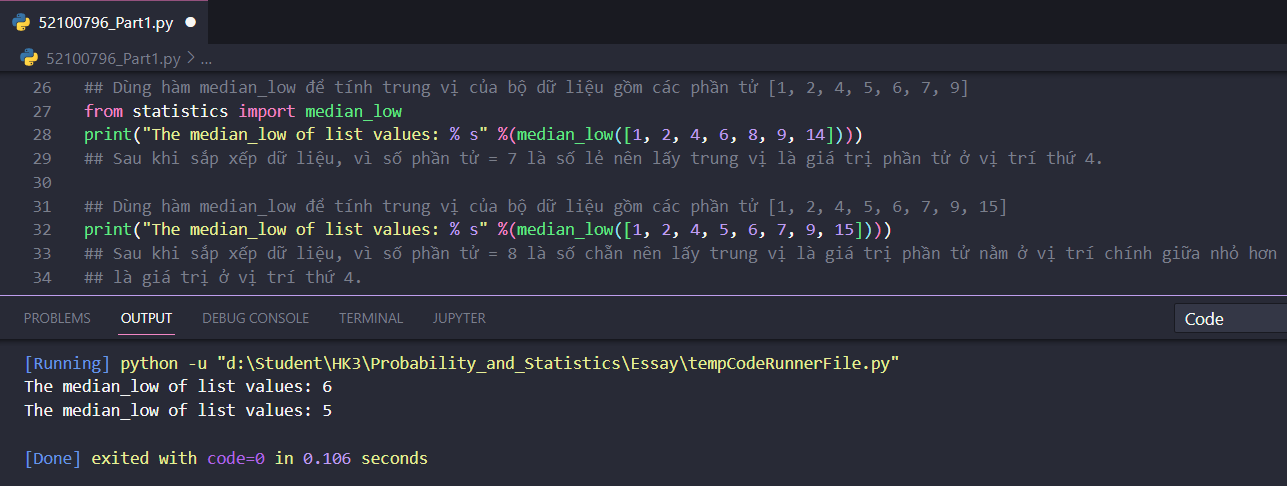
+ Nếu số phần tử của danh sách dữ liệu là số chẵn, hàm median sẽ trả về kết quả là trung bình cộng của giá trị hai phần tử nằm ở vị trí chính giữa của danh sách dữ liệu.

* *Ví dụ:*

**

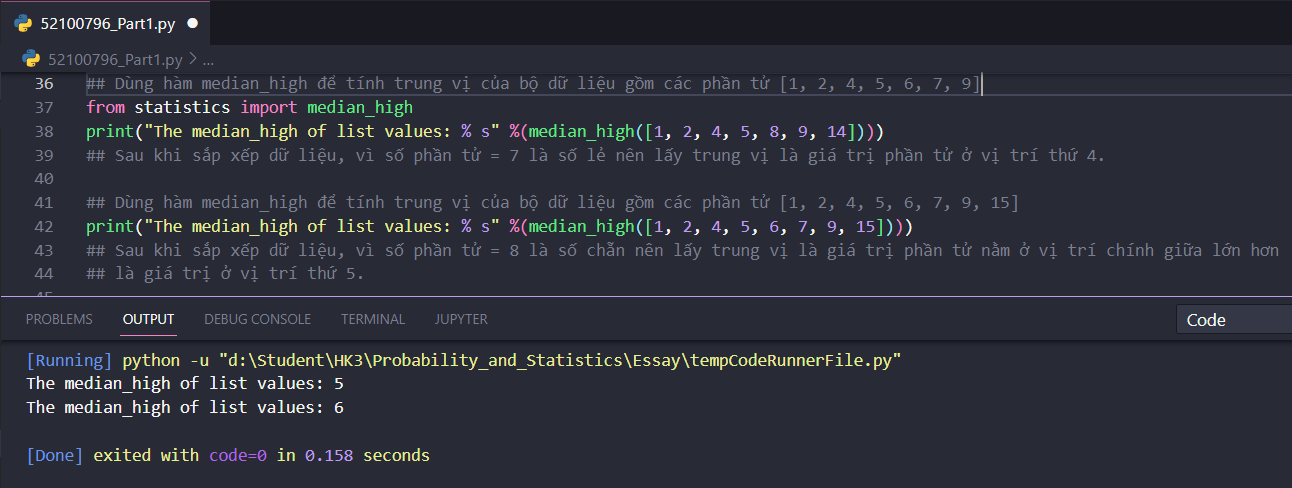
*Hình 5. Sourcecode về ví dụ cho hàm median*

* 1. **Hàm**[[**median\_low()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm meadian\_low(data) cũng giống với hàm median. Tuy nhiên trung vị luôn là một giá trị nằm trong danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* hàm median\_low trả về giá trị chính giữa nằm trong danh sách. Tuy nhiên nếu số phần tử của danh sách dữ liệu là số chẵn thì hàm median\_low trả về giá trị phần tử nằm ở vị trí chính giữa bên trái của danh sách dữ liệu.
* *Ví dụ:*

**

*Hình 6. Sourcecode về ví dụ cho hàm median\_low*

* 1. **Hàm**[[**median\_high()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm median\_high(data) cũng giống với hàm median. Và kết quả trả về của hàm này luôn là một giá trị nằm trong danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* Hàm median\_high trả về giá trị nằm chính giữa danh sách dữ liệu sau khi đã qua bước sắp xếp dữ liệu tăng dần. Nếu số phần tử trong danh sách là số chẵn, hàm median\_high sẽ trả về giá trị phần tử nằm ở vị trí chính giữa bên phải của danh sách dữ liệu.
* *Ví dụ:*

**

*Hình 7. Sourcecode về ví dụ cho hàm median\_high*

* 1. **Hàm**[[**median\_grouped()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm median\_grouped(data, interval = 1) để tính trung vị của bộ dữ liệu liên tục được nhóm lại bằng cách sử dụng phép nội suy.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list, [, interval].
* *Ý nghĩa của kết quả:* kết quả trả về dưới dạng:

Trong đó:

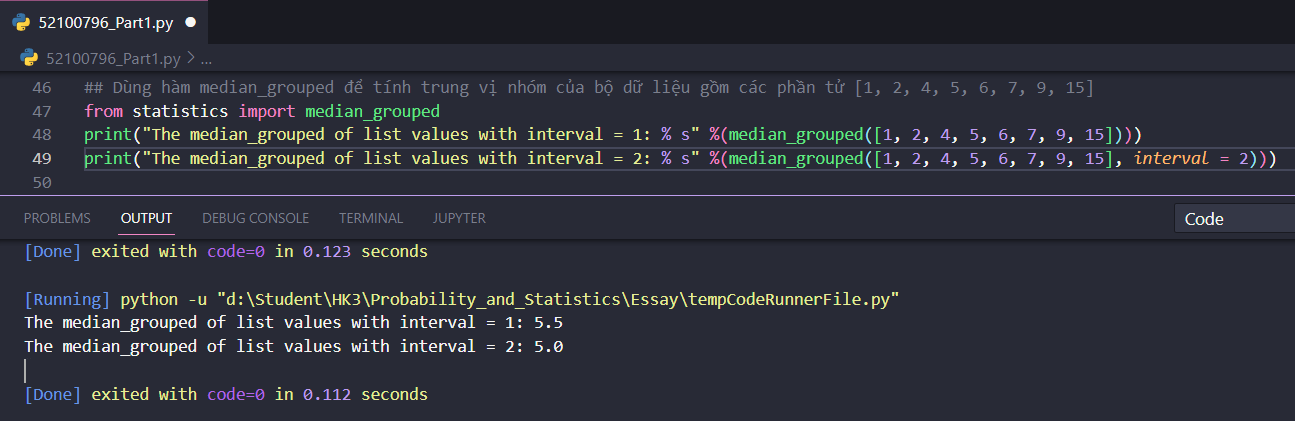
*l*: dưới hạn dưới của lớp trung vị

*n*: số các giá trị.

*cf*: số điểm dữ liệu dưới khoảng trung vị.

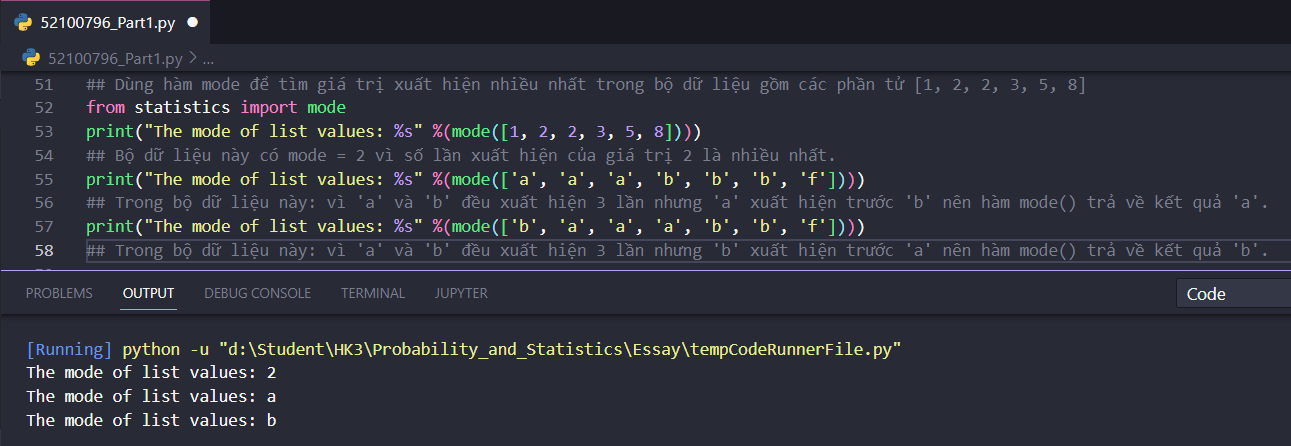
*f:* số điểm dữ liệu trong khoảng trung vị.

* *Ví dụ:*

****

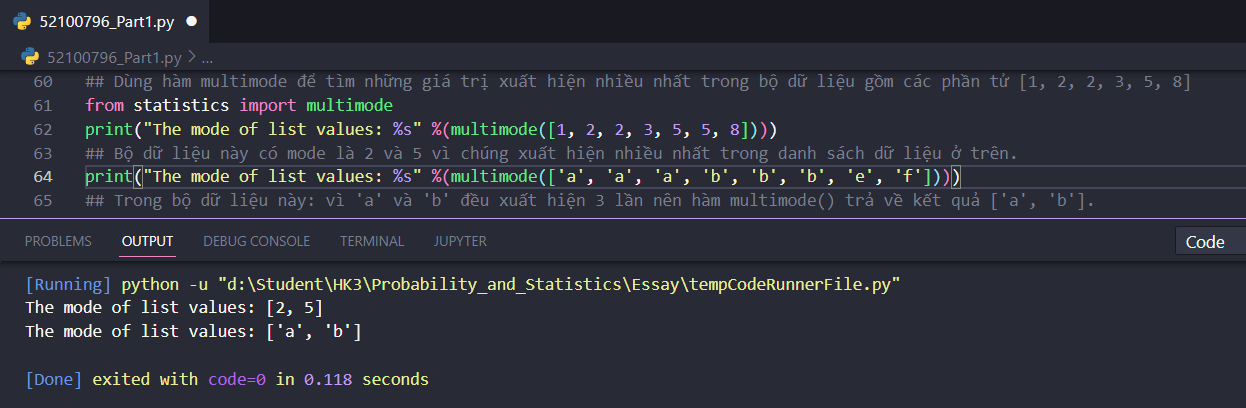
*Hình 8. Sourcecode về ví dụ cho hàm median\_grouped*

* 1. **Hàm**[[**mode()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm mode(data) dùng để tìm giá trị xuất hiện nhiều nhất trong danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* hàm mode trả về giá trị xuất hiện nhiều nhất trong list, có thể là dữ liệu số hoặc dữ liệu khác. Nếu có nhiều hơn một giá trị có số lần xuất hiện bằng nhau, hàm sẽ trả về giá trị xuất hiện trước trong danh sách dữ liệu.
* *Ví dụ:*

**

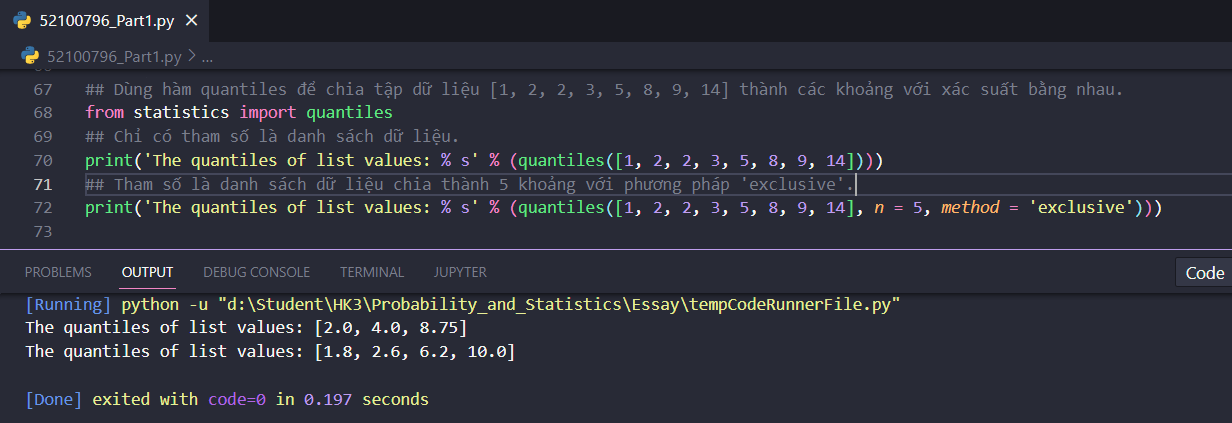
*Hình 9. Sourcecode về ví dụ cho hàm mode*

* 1. **Hàm** [**multi**[**mode()**](https://docs.python.org/3/library/statistics.html#statistics.median)](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**:**
* *Công dụng:* hàm multimode(data) để tìm những giá trị xuất hiện nhiều nhất trong danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list.
* *Ý nghĩa của kết quả:* hàm multimode trả về những giá trị phổ biến nhất trong danh sách dữ liệu, có thể là dữ liệu số hoặc không. Những giá trị này được thể hiện trong một list.
* *Ví dụ:*

**

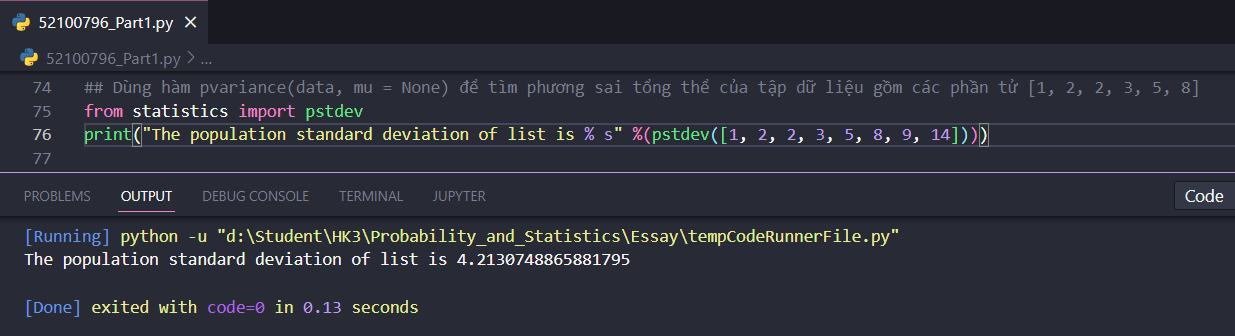
*Hình 10. Sourcecode về ví dụ cho hàm multimode*

* 1. **Hàm** [**quantiles**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**():**
* *Công dụng:* sắp xếp dữ liệu rồi sau đó chia dữ liệu thành các khoảng với xác suất bằng nhau.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list gồm các giá trị [, n] [, method].
* *Ý nghĩa của kết quả:* mặc định n = 4, hàm sẽ tính trung vị của list (Q2), trung vị của các phần tử nhỏ hơn Q2 (Q1), trung vị của các phần tử lớn hơn Q2 (Q3). Sau đó hàm trả về kết quả là 1 list [Q1, Q2, Q3]. Nếu n do người dùng tự đặt, số giá trị trong list trả về là n-1.
* *Ví dụ:*

**

*Hình 11. Sourcecode về ví dụ cho hàm quantiles*

* 1. **Hàm** [**pstdev**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**():**
* *Công dụng:* hàm pstdev(data, mu = None) dùng để tính độ lệch chuẩn chung (tổng thể) của danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list, đối số mu phải là giá trị trung bình của danh sách dữ liệu. Nếu không truyền đối số hoặc để đối số là None, hàm sẽ tự động tính toán giá trị trung bình của danh sách dữ liệu.
* *Ý nghĩa của kết quả:* kết quả trả về độ lệch chuẩn tổng thể (căn bậc 2 của phương sai tổng thể).
* *Ví dụ:*

**

*Hình 12. Sourcecode về ví dụ cho hàm pstdev*

* 1. **Hàm** [**pvariance**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**():**
* *Công dụng:* hàm pvariance(data, mu = None) dùng để tính phương sai chung (tổng thể) của tập hợp. Nó là thước đo để tính độ biến thiên của dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list gồm các giá trị , đối số mu tương tự như hàm pstdev.
* *Ý nghĩa của kết quả:* kết quả trả về phương sai tổng thể là tổng bình phương khoảng cách từ các điểm dữ liệu tới giá trị trung bình chia cho số lượng phần tử trong danh sách dữ liệu. Một phương sai lớn chỉ ra được dữ liệu trải rộng, cách xa so với giá trị trung bình và ngược lại. Theo công thức:

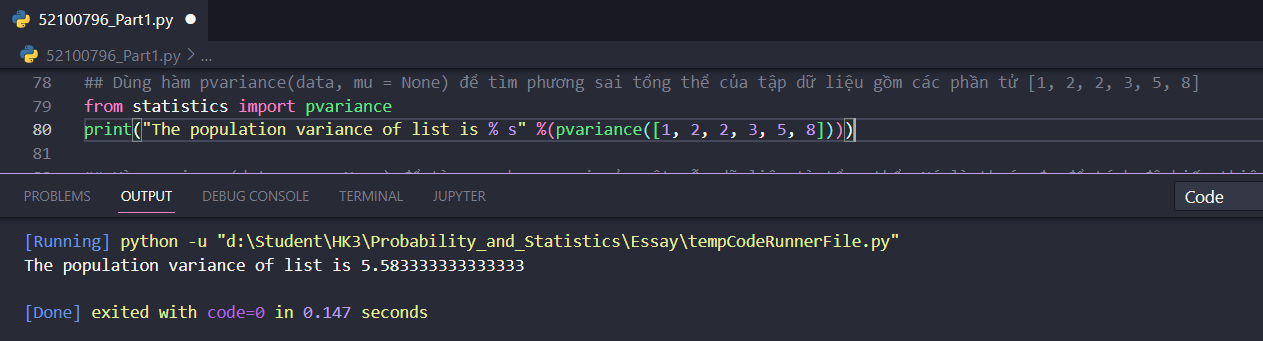
Trong đó:

*n*: số lượng phần tử của tổng thể.

*:* phần tử thứ I trong tổng thể

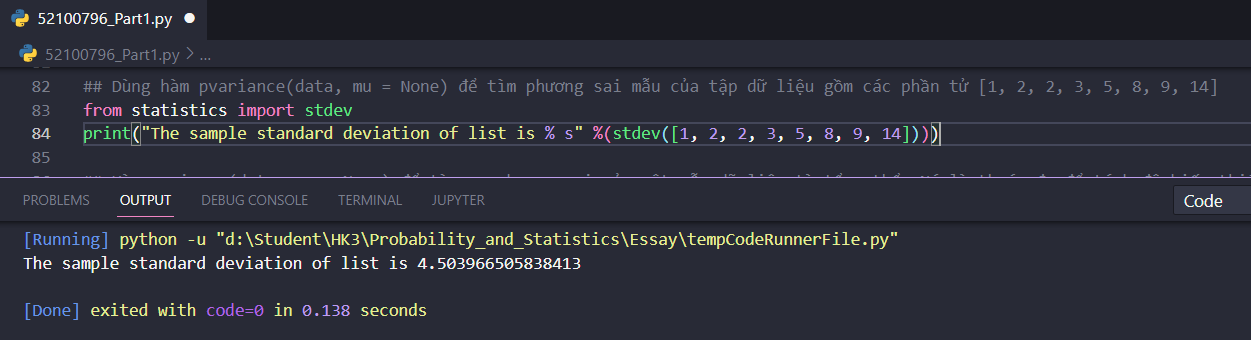
: giá trị trung bình của tổng thể.

* *Ví dụ:*

**

*Hình 13. Sourcecode về ví dụ cho hàm pvariance*

* 1. **Hàm** [**stdev**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**():**
* *Công dụng:* hàm stdev(data, xbar = None) dùng để tính độ lệch chuẩn mẫu của danh sách dữ liệu.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list, đối số xbar nếu được truyền vào thì phải là giá trị trung bình của danh sách dữ liệu. Nếu không truyền hoặc truyền vào đối số là None, gía trị trung bình của dữ liệu mẫu sẽ tự động được tính toán.
* *Ý nghĩa của kết quả:* kết quả trả về độ lệch chuẩn của mẫu dữ liệu (căn bậc 2 của phương sai mẫu).
* *Ví dụ:*

**

*Hình 14. Sourcecode về ví dụ cho hàm stdev*

* 1. **Hàm** [**variance**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean)**():**
* *Công dụng:* hàm variance(data, xbar = None) dùng để tính phương sai mẫu của bộ dữ liệu. Nó cũng tương tự như hàm pvariance.
* *Tham số đầu vào:* danh sách dữ liệu kiểu list gồm các giá trị, đối số xbar tương tự như hàm stdev.
* *Ý nghĩa của kết quả:* kết quả trả về phương sai của mẫu dữ liệu là tổng bình phương khoảng cách từ các điểm đến giá trị trung bình cảu mẫu dữ liệu chia cho số phần tử của dữ liệu mẫu.Ý nghĩa của phương sai mẫu cũng giống như phương sai của tổng thể. Theo công thức:

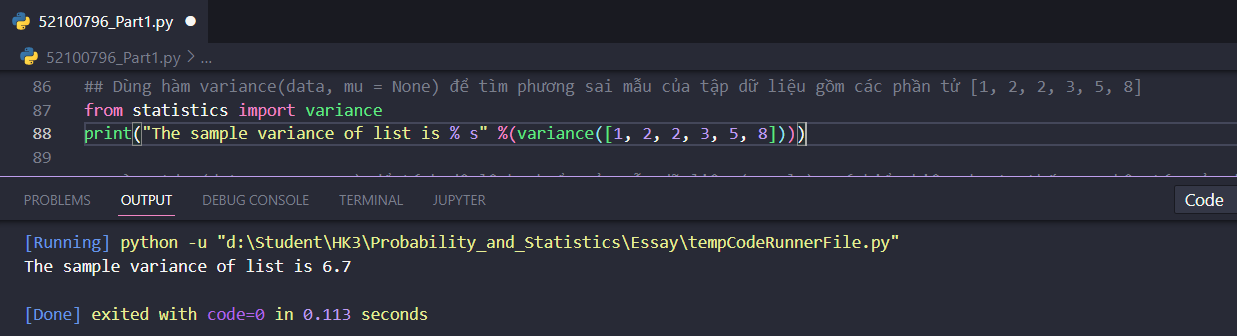
Trong đó:

*n*: số lượng phần tử trong danh sách dữ liệu mẫu.

: phần tử thứ i trong danh sách dữ liệu mẫu

: giá trị trung bình của danh sách dữ liệu mẫu.

* *Ví dụ:*

**

*Hình 15. Sourcecode về ví dụ cho hàm variance*

* 1. **Hàm covariance():**
* *Công dụng:* hàm covariance(x, y) dùng để tìm hiệp phương sai của hai danh sách dữ liệu x và y. Nó cũng là thước đo của sự thay đổi chung của danh sách dữ liệu x và y.
* *Tham số đầu vào:* hai danh sách dữ liệu kiểu list, độ dài của hai danh sách phải bằng nhau và phải lớn hơn 2.
* *Ý nghĩa của kết quả:* kết quả trả về hiệp phương sai của hai danh sách dữ liệu x và y được tính theo công thức:

Trong đó:

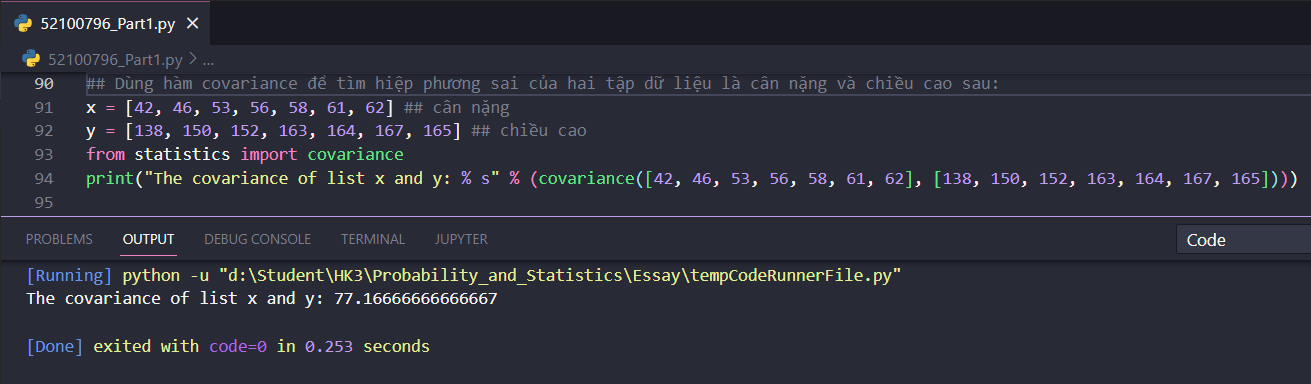
*n*: số lượng phần tử của bộ dữ liệu.

: giá trị trung bình của danh sách dữ liệu x.

: giá trị trung bình của danh sách dữ liệu y.

: phần tử thứ i trong danh sách dữ liệu x, y .

* *Ví dụ:*

**

*Hình 16. Sourcecode về ví dụ cho hàm covariance*

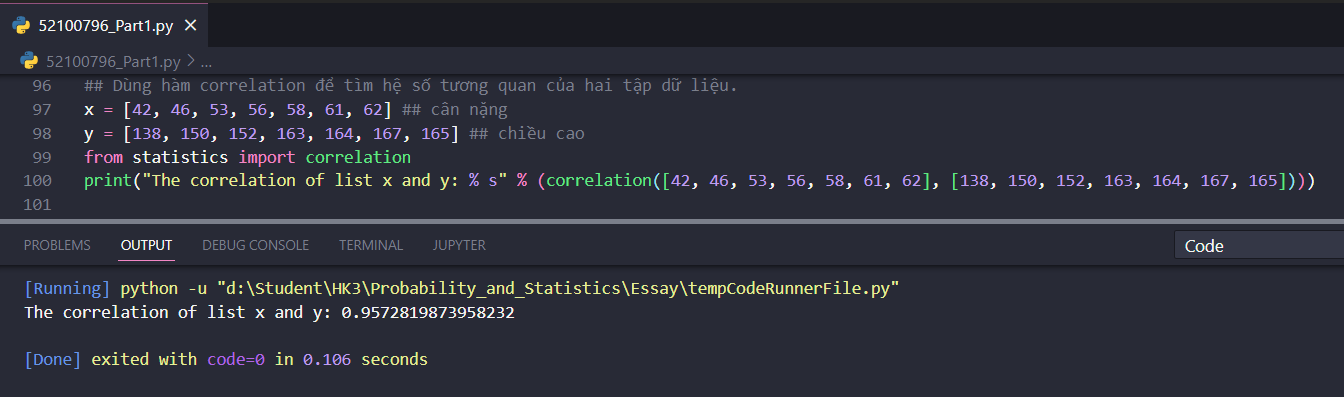
* 1. **Hàm correlation():**
* *Công dụng:* hàm correlation(x, y) dùng để tìm hệ số tương quan của hai tập dữ liệu x và y.
* *Tham số đầu vào:* hai danh sách dữ liệu kiểu list, độ dài của hai danh sách phải bằng nhau và phải lớn hơn 2.
* *Ý nghĩa của kết quả:* kết quả trả về hệ số tương quan Pearson. Hệ số này có giá trị nằm trong khoảng từ -1 đến 1.

+ Nếu hệ số tương quan hướng dần tới 1: tương quan hai danh sách dữ liệu thuận.

+ Nếu hệ số tương quan hướng dần tới -1: tương quan hai danh sách dữ liệu nghịch.

+ Nếu hệ số tương quan hướng dần đến 0: hai danh sách dữ liệu không có sự tương quan với nhau.

* *Ví dụ:*

**

*Hình 17. Sourcecode về ví dụ cho hàm correlation*

* 1. **Hàm linear\_regression():**
* *Công dụng:* hàm linear\_regression(x, y) dùng để trả về độ dốc (slope) và hệ số chặn (intercept) là hai hệ số của phương trình hồi quy tuyến tính.
* *Tham số đầu vào:* hai danh sách dữ liệu kiểu list, độ dài của hai danh sách phải bằng nhau và phải lớn hơn 2.
* *Ý nghĩa của kết quả:* hàm linear\_regression trả về độ dốc và hệ số chặn để ra được phương trình hồi quy tuyến tính. Phương trình này có dạng:

y = slope \* x + intercept + noise

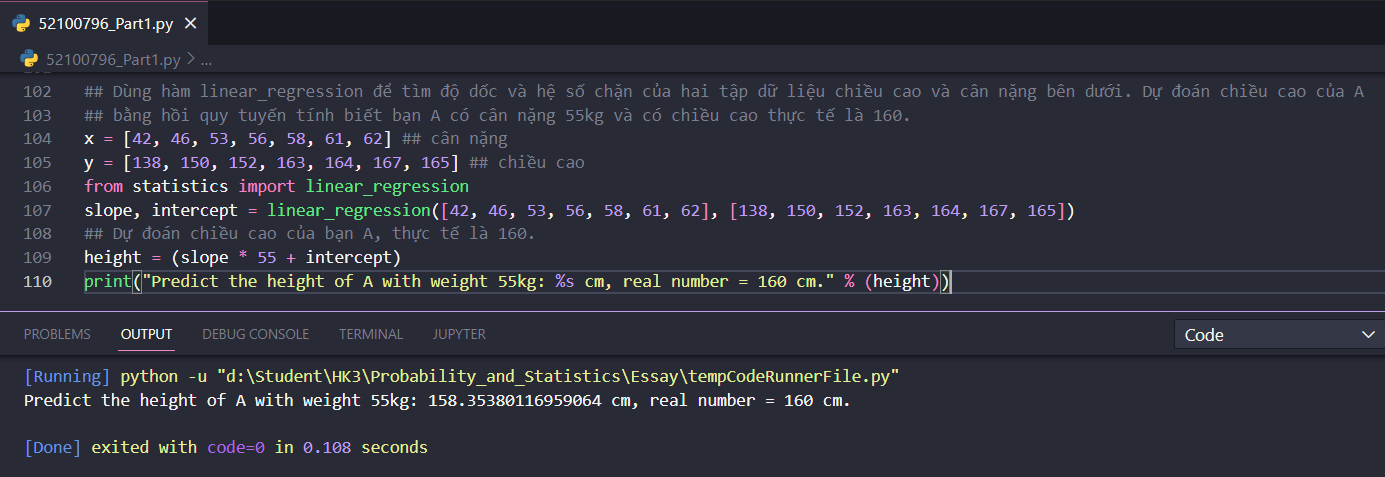
Trong đó:

+ slope là độ dốc.

+ intercept là hệ số chặn.

+ noise là sự biến thiên của dữ liệu so với với thực tế.

* *Ví dụ:*

**

*Hình 18. Sourcecode về ví dụ cho hàm linear\_regression*

1. **PHƯƠNG ÁN KHẮC PHỤC, BIỆN PHÁP PHÒNG TRÁNH**
   1. **Vấn đề:**

Ngày nay, việc chụp ảnh để lưu giữ những khoảnh khắc dường như dã trở nên không còn xa lạ với cuộc sống thường ngày của chúng ta. Tuy nhiên trong thực tế, những bức ảnh thường có thể bị tối màu hoặc quá sáng do những nguyên nhân như góc chụp, điều kiện sáng,.. Vì thế trong lĩnh vực xử lí ảnh số, có một số thuật toán được ra đời, trong đó thuật toán cân bằng Histogram, là thuật toán có thể cải thiện được chất lượng của các bức ảnh.

* 1. **Những ràng buộc và điều kiện của thuật toán:**

Khi chọn những hình ảnh có độ sâu màu thấp như hình ảnh có thang màu xám 8 bit sẽ tạo ra những hiệu ứng không mong muốn. Chính vì vậy, phương pháp này hoạt dộng tốt nhất trên những hình ảnh có độ sâu màu cao hơn như hình ảnh có thang màu xám 16 bit hoặc dữ liệu liên tục.

Phương pháp cân bằng Histogram hữu ích trong các ảnh nền vừa sáng hoặc vừa tối. Chẳng hạn như chụp X-quang, tuy nhiên phương pháp này mang nhược điểm là tăng độ tương phản của những thành phần gây nhiễu xung quanh. Điều này có thể làm giảm tín hiệu khả dụng của bức ảnh.

* 1. **Thuật toán cân bằng Histogram**

B1: Hình ảnh sẽ được biểu thị dưới dạng ma trận vuông. Mỗi ô trong bảng sẽ thể hiện mức xám tương ứng với vị trí của hình ảnh.

B2: Thống kê số lượng pixel cho từng mức sáng.

B3: Từ bảng thống kê ta lập ra biểu đồ gồm năm cột:

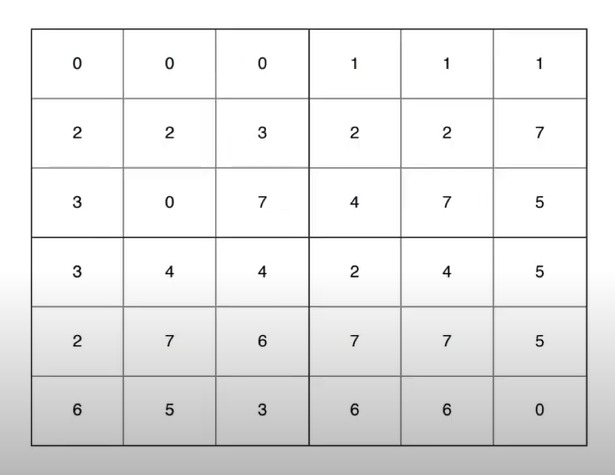
* Cột thứ nhất: Các điểm xám xuất hiện trong ma trận, sắp xếp tăng dần. Ta gọi là hàm g(i).
* Cột thứ hai: Số lần xuất hiện của từng mức xám tương ứng. h(g(i))
* Cột thứ ba: Số lần xuất hiện tích lũy của các mức xám. c(g(i))
* Cột thứ tư: Tần suất tích lũy cho từng mức xám, là số lần xuất hiện tích lũy chia cho tổng số mức xám. f(g(i))
* Cột thứ năm: giá trị mức xám mới. Tính theo công thức:

Trong quá trình tính toán. Nếu thấy newpixel có giá trị bằng với giá trị mức xám hiện tại đang xét. Ta không cần tính mà ghi lại giá trị mức xám ban đầu cho đến mức xám lớn nhất. Đặc biệt trong xử lí ảnh số, chúng ta dùng công thức:

B4: Thay thế giá trị mức xám của ma trận bằng các newpixel tương ứng từ biểu đồ đã tính.

* 1. **Ví dụ**

Ta có một bức ảnh được đơn giản hóa thành ma trận với mức sáng từ 0 đến 7 pixel.



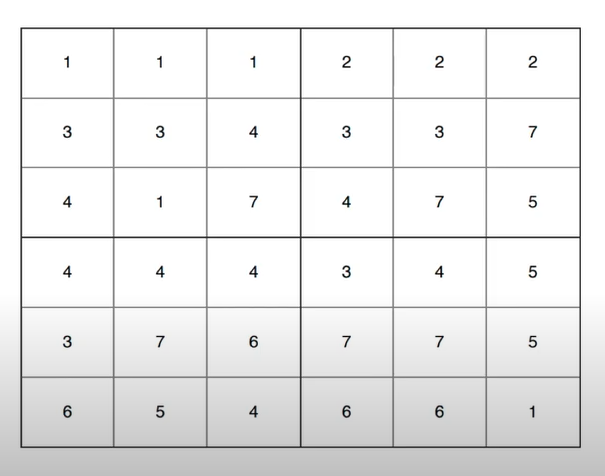
*Hình 19. Ma trận của bức ảnh có mức sáng từ 0 đến 7 pixel*

Tiếp theo, chúng ta thực hiện tuần tự các bước theo thuật toán cân bằng Histogram.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| g(i) | h(g(i)) | c(g(i)) | f(g(i)) | newpixel |
| 0 | 5 | 5 | 5/36 | 1 |
| 1 | 3 | 8 | 8/36 | 2 |
| 2 | 6 | 14 | 14/36 | 3 |
| 3 | 4 | 18 | 18/36 | 4 |
| 4 | 4 | 22 | 22/36 | 4 |
| 5 | 4 | 26 | 26/36 | 5 |
| 6 | 4 | 30 | 30/36 | 6 |
| 7 | 6 | 36 | 1 | 7 |

*Bảng 1. Chuẩn hóa các giá trị pixel, tăng độ tương phản.*

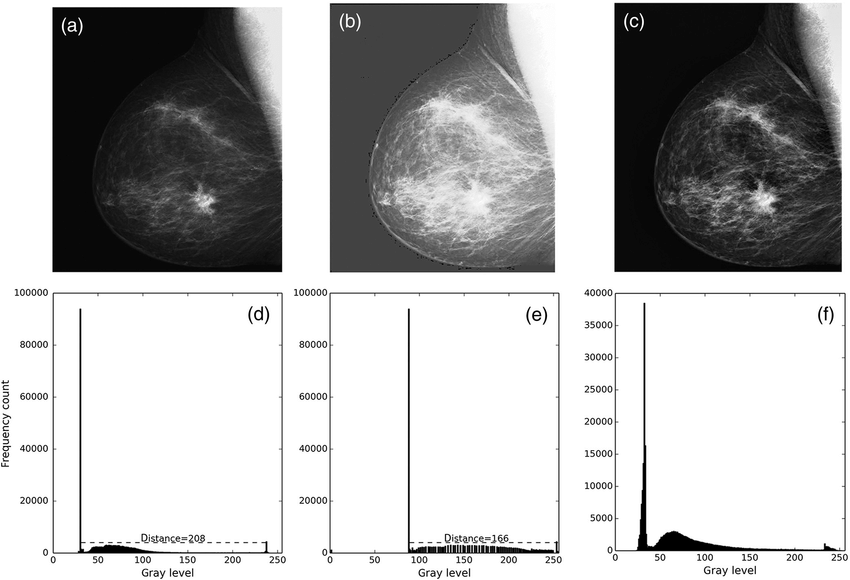
Cuối cùng, thay thế các mức xám cũ bằng giá trị newpixel. Ta được ma trận mới là ma trận mới. Ma trận này đại diện cho sự cân bằng của thuật toán cân bằng Histogram.



*Hình 20. Ma trận của bức ảnh sau khi trải qua thuật toán cân bằng Histogram*

* 1. **Nhận xét, đánh giá**

Như đã trình bày ở phần 2.2, thuật toán này hữu ích trong các bức ảnh có nền vừa sáng hoặc vừa tối. Giúp chúng ta có cái nhìn rõ hơn về cấu trúc xương trong hình ảnh chụp X-quang. Thuật toán này cũng rất đơn giản và dễ thực hiện. Tuy nhiên, khi thực hiện thuật toán thì những giá trị gây nhiễu cũng sẽ được tăng độ tương phản lên. Điều này làm giảm đi những tín hiệu thông tin cần đạt được. Chẳng hạn như hình sau:



*Hình 21. Thuật toán cân bằng Histogram khi gặp nhiều dữ liệu nhiễu*

1. **TRIỂN KHAI WEBSITE PHÒNG TRÁNH CÁC LỖI BẢO MẬT THƯỜNG GẶP**
   1. **Thư viện statistics trong Python**

Để sử dụng thư viện statistics. Ta dùng lệnh *import statistics*. Ngoài ra, trong bài tiểu luận này, em sử dụng câu lệnh *from statistics import <method>* để tối ưu các hàm trong thư viện.

Đối với mỗi hàm, cách giải thích và công thức tính toán đã được trình bày ở phần 1. Vì thế em tập trung vào thuật toán cân bằng Histogram.

* 1. **Thuật toán cân bằng Histogram**

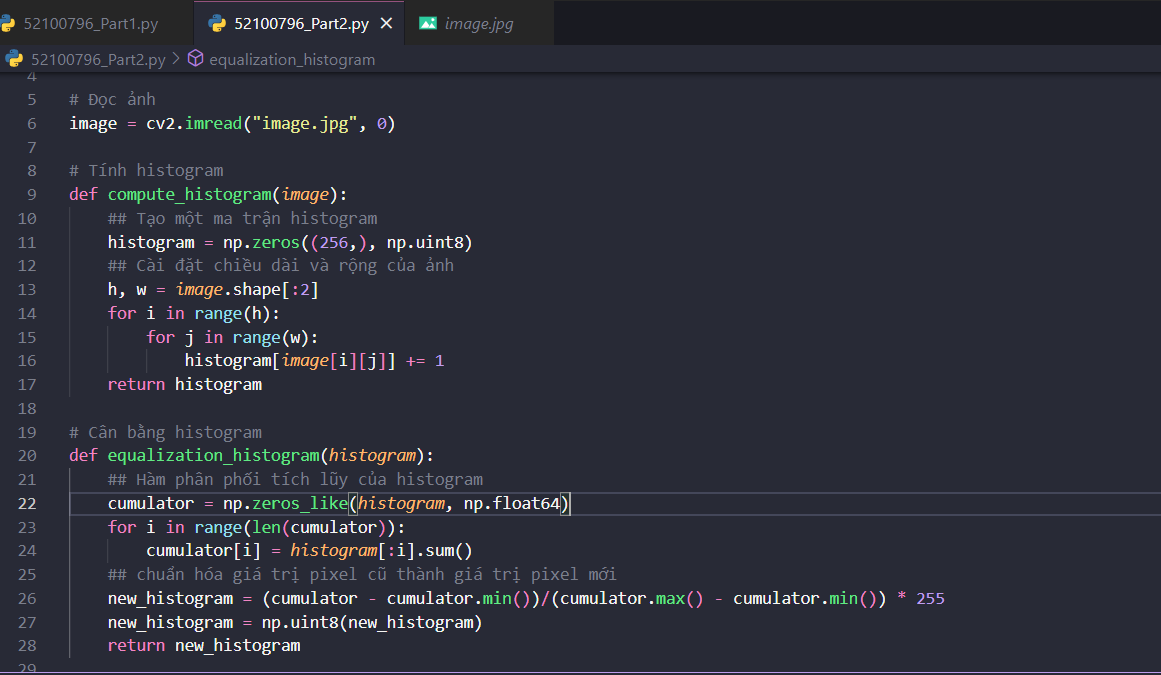
Trong phần này, chúng ta cần sử dụng những thư viện numpy, matplotlib và thư viện OpenCV. Thư viện numpy giúp chúng ta xử lí các phần về ma trận được tạo ra từ hình ảnh. Thư viện matplotlib để ta vẽ đồ thị và openCV khai báo thư viện cv2 giúp chúng ta xử lí bức ảnh.



*Hình 22. Ảnh trước khi xử lí*

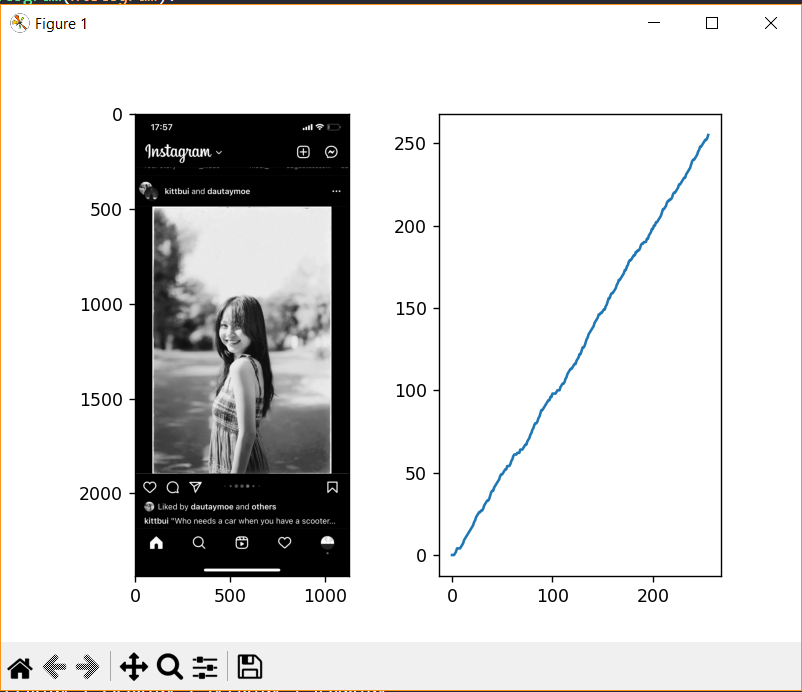
Tiếp theo, cần đọc hình ảnh vào với hình ảnh được nằm chung với thư mục chứa file 52100796\_Part2.py. Sử dụng câu lệnh *cv2.imread(<image>, 0)*.

Kế đến, chúng ta có hai hàm để xử lí ma trận pixel của ảnh đó là *compute\_histogram(image)* và *equalization\_histogram(histogram)*. Đối với hàm *compute\_histogram*, tham số đầu vào là hình ảnh. Sau đó hàm này trả về histogram các giá trị pixel của ảnh. Những pixel này nằm trong khoảng từ 1 đến 255. Còn đối với hàm *equalization\_histogram*, hàm này sẽ giúp chúng ta tính toán để biến đổi những giá trị pixel cũ thành giá trị pixel mới để cải thiện độ tương phản của ảnh.



*Hình 23. Lệnh đọc ảnh và hàm tính toán cho việc xử lí ảnh*

Cuối cùng, chúng ta xem kết quả hình ảnh mới cũng như đồ thị bằng câu lệnh *imshow(<image\_processed>, cmap = ‘gray’)* để tạo ra ảnh xám và thể hiện đồ thị của bức ảnh đó với lệnh *plt.show()*. Kết quả xử lí ảnh được thể hiện trong hình bên dưới:



*Hình 24. Hình ảnh được xử lí bằng thuật toán cân bằng histogram*

**DANH MỤC TÀI LIỆU THAM KHẢO VÀ CÁC TRÍCH DẪN**

[1] *Mathematical statistics functions -* [*https://docs.python.org/3/library/statistics.html*](https://docs.python.org/3/library/statistics.html)

[2] *Xử lí ảnh: Thuật toán cân bằng histogram -* [*https://viblo.asia/p/xu-li-anh-thuat-toan-can-bang-histogram-anh-GrLZDOogKk0*](https://viblo.asia/p/xu-li-anh-thuat-toan-can-bang-histogram-anh-GrLZDOogKk0)

[3] *Statistical Functions in Python -* [*Statistical Functions in Python | Set 1 (Averages and Measure of Central Location) - GeeksforGeeks*](https://www.geeksforgeeks.org/statistical-functions-python-set-1averages-measure-central-location/)

[4] *OpenCV Python Tutorial -* [*https://www.geeksforgeeks.org/opencv-python-tutorial/*](https://www.geeksforgeeks.org/opencv-python-tutorial/)

[5] *Histogram equalization - https://en.wikipedia.org/wiki/Histogram\_equalization*

--Hết--