TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ MÔN XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG CHO CNTT**

*Người hướng dẫn*: **GV. LÊ ANH TUẤN**

*Người thực hiện*: **TRẦN THỊ VẸN - 53200674**

Lớp **: 21050301**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ MÔN XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG CHO CNTT**

*Người hướng dẫn*: **GV. LÊ ANH TUẤN**

*Người thực hiện*: **TRẦN THỊ VẸN - 53200674**

Lớp **: 21050301**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành gửi lời cảm ơn và sự tri ân sâu sắc đến với các thầy cô, giảng viên của khoa Công nghệ thông tin nói chung và các thầy cô môn Xác suất thống kê nói riêng. Trong suốt quá trình học tập và rèn luyện, em đã nhận được rất nhiều sự giúp đỡ tận tình, sự quan tâm, chăm sóc của các thầy cô.

Ngoài ra, em còn được thầy Lê Anh Tuấn và cô Nguyễn Thị Huỳnh Trâm, thầy cô đã dạy bảo những kiến thức, phương pháp mới về toán hay ho và thú vị, thầy cô còn giúp sinh viên có được nhiều niềm vui trong việc học và cảm thấy thoải mái, … Em xin cảm ơn các thầy cô rất nhiều trong suốt quá trình học tập này ạ!!!

Bởi lượng kiến thức của em còn hạn hẹp và gặp nhiều vấn đề trong quá trình học nên báo cáo giữa kỳ này sẽ còn nhiều thiếu sót và cần được học hỏi thêm. Em rất mong em sẽ nhận được sự góp ý của quý thầy cô về bài báo cáo này của em để em rút kinh nghiệm trong những môn học sắp tới. Cuối cùng, em xin chân thành cảm ơn quý thầy cô.

TP Hồ Chí Minh, ngày 01 tháng 11 năm 2022

Sinh viên: Trần Thị Vẹn

**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 GV. Lê Anh Tuấn. 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 01 tháng 11 năm 2022*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Thị Vẹn*

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

Xác suất và thống kê toán học là một ngành khoa học đang giữ vị trí quan trọng trong các lĩnh vực ứng dụng rộng rãi và phong phú của đời sống con người. Cùng với sự phát triển mạnh mẽ của khoa học và công nghệ, nhu cầu hiểu biết và sử dụng các công cụ ngẫu nhiên trong phân tích và xử lý thông tin ngày càng trở nên đặc biệt cần thiết.

Báo cáo giữa kì Xác suất thống kê này bao gồm các kiến thức về các chức năng trong thư viện Thống kê và các phép tính toán trung bình, trung vị, yếu vị, độ lệch chuẩn, phương sai và thuật toán cân bằng biểu đồ để xử lý hình ảnh điều chỉnh độ tương phản của ảnh bằng cách sử dụng biểu đồ của hình ảnh…

Trong bài báo cáo này tôi sẽ áp dụng các kiến thức mà giảng viên đã cung cấp trong quá trình giảng dạy để giải quyết các câu hỏi cũng như vấn đề đã được đặt ra trong đề báo cáo . Thêm vào đó, sử dụng các hình ảnh được code trong Visual Studio Code để minh họa các thuật toán trong quá trình làm bài báo cáo giữa kỳ.

Bài báo cáo gồm 2 phần chia làm 3 chapter nên đòi hỏi sinh viên cần phải vững nắm vững kiến thức để có thể suy luận và kết hợp lại với nhau nên đôi khi sẽ khó tránh khỏi sai sót nhưng tôi sẽ cố gắng hoàn thành thật tốt đề tài này thật tốt.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc387692905)

[LỜI CAM KẾT ii](#_Toc387692906)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc387692906)

[TÓM TẮT i](#_Toc387692907)v

[MỤC LỤC 1](#_Toc387692908)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc387692909)

PHẦN 1: CÁC HÀM - CHỨC NĂNG CỦA THƯ VIỆN THỐNG KÊ ………….……5

[1.1 Cơ sở lý thuyết chung về thư viện Thống kê (Statistics library) 5](#_Toc387692912)

[1.1.1 Thống kê mô tả (Descriptive Statistics) 5](#_Toc387692912)

[1.1.2 Tổng quan thư viện thống kê trong Python 5](#_Toc387692912)

[1.2 Thống kê mô tả (Descriptive Statistics) 5](#_Toc387692912)

[1.2.1 Hàm mean() - hàm tính giá trị trung bình 5](#_Toc387692912)

[1.2.2 Hàm fmean() - hàm tính giá trị trung bình](#_Toc387692912) 6

[1.2.3 Hàm geometric() – hàm tính giá trị trung bình nhân (hình học) 7](#_Toc387692912)

[1.2.4 Hàm harmonic() – hàm tính giá trị trung bình điều hòa 8](#_Toc387692912)

[1.2.5 Hàm median() – hàm tính giá trị trung vị 9](#_Toc387692912)

[1.2.6 Hàm median\_low() – hàm tính giá trị trung vị thấp 1](#_Toc387692912)0

[1.2.7 Hàm median\_high() – hàm tính giá trị trung vị cao 1](#_Toc387692912)1

[1.2.8 Hàm median\_grouped() 1](#_Toc387692912)2

[1.2.9 Hàm mode() – trả về giá trị xuất hiện nhiều nhất 1](#_Toc387692912)3

[1.2.10 Hàm multimode() – hàm trả về danh sách xuất hiện thường xuyên 1](#_Toc387692912)3

[1.2.11 Hàm quantiles() – hàm trả về những điểm cắt 1](#_Toc387692912)4

[1.2.12 Hàm pstdev() – hàm tính độ lệch chuẩn tổng thể 1](#_Toc387692912)5

[1.2.13 Hàm pvariance() – tính phương sai tổng thể 1](#_Toc387692912)6

[1.2.14 Hàm stdev() – hàm tính độ lệch chuẩn mẫu 1](#_Toc387692912)7

[1.2.15 Hàm variance() – tính phương sai mẫu 18](#_Toc387692912)

[1.2.16 Hàm covariance() – tính phương sai mẫu 19](#_Toc387692912)

[1.2.17 Hàm correlation() – hàm tính hệ số tương quan của Person 20](#_Toc387692912)

[1.2.18 Hàm linear\_regression() – phương trình hồi quy tuyến tính 21](#_Toc387692912)

[PHẦN 2: THUẬT TOÁN CÂN BẰNG HISTOGRAM – XỬ LÝ ẢNH 23](#_Toc387692917)

[2.1 Cơ sở lý thuyết chung về thuật toán cân bằng histogram 23](#_Toc387692918)

[2.1.1 Lý thuyết về xử lý ảnh và histogram 23](#_Toc387692912)

[2.1.2 Lý thuyết về cân bằng histogram 24](#_Toc387692912)

[2.2.1.1 Cân bằng histogram 24](#_Toc387692912)

[2.2.1.2 Ví dụ, nhận xét, phân tích và đánh giá 24](#_Toc387692912)

[2.2 Thuật toán Histogram và các bước thực hiện 25](#_Toc387692918)

[PHẦN 3: THỰC THI (IMPLEMENTATION) 25](#_Toc387692920)

[3.1 Giải thích và hướng dẫn chạy code của thuật toán 25](#_Toc387692921)

[3.2 Kết quả thử nghiệm 27](#_Toc387692922)

DANH MỤC HÌNH VẼ, ĐỒ THỊ VÀ TÀI LIỆU THAM KHẢO

**DANH MỤC HÌNH**

[Hình 1: Input của hàm mean 7](#_Toc118752536)

[Hình 2: Output của hàm mean 7](#_Toc118752537)

[Hình 3: Input của hàm fmean 8](#_Toc118752538)

[Hình 4: Output của hàm fmean 8](#_Toc118752539)

[Hình 5: Input của hàm trung bình nhân 9](#_Toc118752540)

[Hình 6: Output của hàm trung bình nhân 9](#_Toc118752541)

[Hình 7: Input của hàm harmonic 9](#_Toc118752542)

[Hình 8: Output của hàm harmonic 10](#_Toc118752543)

[Hình 9: Input của hàm harmonic có trọng số 10](#_Toc118752544)

[Hình 10:Output của hàm harmonic có trọng số 10](#_Toc118752545)

[Hình 11: Input của hàm median 11](#_Toc118752546)

[Hình 12: Output của hàm median 11](#_Toc118752547)

[Hình 13: Input của hàm median\_low 12](#_Toc118752548)

[Hình 14: Output của meadian\_low 12](#_Toc118752549)

[Hình 15: Input của hàm median\_high 12](#_Toc118752550)

[Hình 16: Output của hàm median\_high 13](#_Toc118752551)

[Hình 17: Input của hàm median\_grouped 13](#_Toc118752552)

[Hình 18: Output của hàm median\_high 13](#_Toc118752553)

[Hình 19: Input của hàm mode 14](#_Toc118752554)

[Hình 20: Output của hàm mode 14](#_Toc118752555)

[Hình 21: Input của hàm multimode 15](#_Toc118752556)

[Hình 22: Output của hàm multimode 15](#_Toc118752557)

[Hình 23: Input của hàm quantiles 16](#_Toc118752558)

[Hình 24:Output của hàm quantiles 16](#_Toc118752559)

[Hình 25: Output của hàm quantiles 16](#_Toc118752560)

[Hình 26: Input của hàm pstdev 17](#_Toc118752561)

[Hình 27: Output của hàm pstdev 17](#_Toc118752562)

[Hình 28: Input của hàm Pvariance 18](#_Toc118752563)

[Hình 29: Output của hàm Pvariance 18](#_Toc118752564)

[Hình 30: Input của hàm stdev 19](#_Toc118752565)

[Hình 31: Output của hàm stdev 19](#_Toc118752566)

[Hình 32: Input của hàm Variance 20](#_Toc118752567)

[Hình 33: Output của hàm Variance 20](#_Toc118752568)

[Hình 34: Input của hàm covariance 21](#_Toc118752569)

[Hình 35: Output của hàm covariance 21](#_Toc118752570)

[Hình 36: Output của hàm covariance 21](#_Toc118752571)

[Hình 37: Input của hàm correlation 22](#_Toc118752572)

[Hình 38: Output của hàm correlation 22](#_Toc118752573)

[Hình 39: Output của hàm correlation 22](#_Toc118752574)

[Hình 40: Input của hàm hồi quy tuyến tính – linear\_regression 23](#_Toc118752575)

[Hình 41: Output của hàm hồi quy tuyến tính – linear\_regression 23](#_Toc118752576)

[Hình 42: Output của hàm hồi quy tuyến tính – linear\_regression 23](#_Toc118752577)

[Hình 43: Biểu đồ phân bố pixels 24](#_Toc118752578)

[Hình 44: Biểu đồ cho thấy độ sáng cũng như histogram của ảnh 25](#_Toc118752579)

[Hình 45: Ảnh xám trắng đen bị mờ trước và sau cân bằng 28](#_Toc118752580)

[Hình 46: Ảnh hội nghị mờ được chỉnh rõ hơn 29](#_Toc118752581)

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Fundamentals of Descriptive Statistics, September 1, 1997, by Routledge

[2] “Digital Image Processing”, 2002, R. C. Gonzalez and R. E. Woods

PHẦN 1: CÁC HÀM - CHỨC NĂNG CỦA THƯ VIỆN THỐNG KÊ

**Khảo sát tất cả các chức năng của thư viện Thống kê trong Python. Tương ứng, mỗi hàm, chức năng cần được giải thích về cách sử dụng, các tham số đầu vào, ý nghĩuza của kết quả trả về và code ví dụ.**

**https://docs.python.org/3/library/stosystem.html**

* 1. Cơ sở lý thuyết chung về thư viện Thống kê (Statistics library):

Trước khi tìm hiểu về thư viện Thống kê trong Python, chúng ta cùng đi tìm hiểu về khái niệm của Thống kê mô tả và hình ảnh bao gồm các giá trị của thống kê mô tả.

* + 1. **Thống kê mô tả (Descriptive Statistics):**

Thống kê mô tả là các hệ số được mô tả ngắn gọn hay tóm tắt một tập dữ liệu nhất định, cũng có thể là đại diện cho toàn bộ hoặc một mẫu của một tổng thể. Thống kê mô tả được chia thành:

* Đo lường xu hướng tập trung: giá trị trung bình (mean), trung vị (median) và yếu via (mode).
* Đo lường biến động: độ lệch chuẩn (standard deviation), phương sai (variance), giá trị nhỏ nhất (min) và giá trị lớn nhất (max), độ nhọn (kurtosis) và độ lệch (skewness).
  + 1. **Thư viện thống kê trong Python:**

Thư viện thống kê của Python tích hợp sẵn có một số lượng tương đối các hàm thống kê quan trọng nhất trong thống kê mô tả. Bạn có thể sử dụng thư viện này để tính toán các thống kê toán học của dữ liệu số (giá trị thực). Ngoài ra, bạn chỉ nên dung thư viện này nếu tập dữ liệu của bạn không quá lớn hoặc nếu bạn không thể dựa vào hay tin tưởng việc chèn vào các thư viện khác để tính thống kê. Khi sử dụng thư viện bạn cần phải chèn câu lệnh: **import statistics.**Thư viện thống kê có version mới nhất trong Python là version 3.4.

* 1. **Các hàm trong thư viện thống kê của Python:**
     1. **Hàm mean() - hàm tính giá trị trung bình:**

Statistics.mean() là hàm được sử dụng để tính giá trị trung bình cộng của tập dữ liệu. Về cú pháp sử dụng của hàm này là: **statistics.mean(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator). Ngoài ra,** nếu data trống nó trả về StatisticsError. Kết quả trả về của hàm phụ thuộc vào kiểu dữ liệu đầu vào nếu nhập vào danh sách số nguyên trả về số nguyên, danh sách số thực sẽ trả về số thực.

Ví dụ: Cho một danh sách gồm 6 giá trị: Mean1 = [1, 2, 0, 9, 10, 8]

Text

Description automatically generated

Hình : Input của hàm mean

Text

Description automatically generated

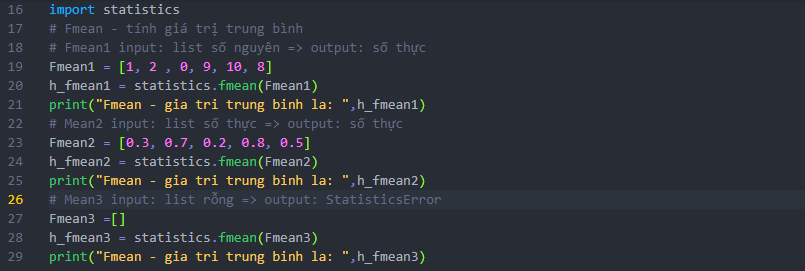
Hình 2: Output của hàm mean

**1.2.2 Hàm fmean() – hàm tính giá trị trung bình:**

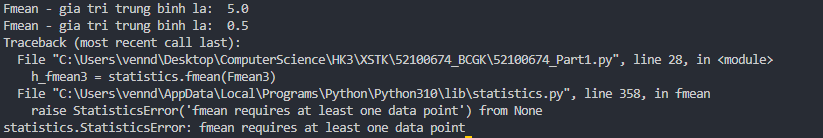
Hàm statistics.fmean() cũng là hàm được sử dụng để tính giá trị trung bình cộng của tập dữ liệu giống như mean(). Cú pháp sử dụng là: **statistics.fmean(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator). Ngoài ra,** nếu data trống nó trả về StatisticsError, hàm luôn trả về giá trị thuộc kiểu float và chạy nhanh hơn hàm mean().

Ví dụ: Cho một danh sách gồm 6 giá trị: Fmean1 = [1, 2, 0, 9, 10, 8]

Ta có công thức tính trung bình cộng:



Hình 3: Input của hàm fmean



Hình 4: Output của hàm fmean

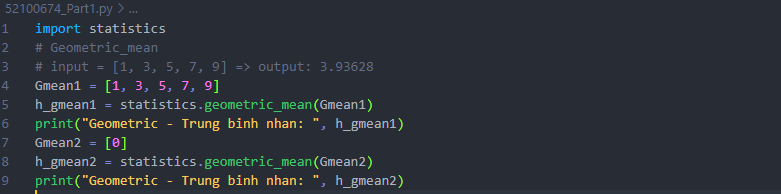
Ngoài ra, đối python version 3.11 còn hỗ trợ hàm fmean() kèm với trọng số. Cú pháp sử dụng là **statistics.fmean(data,weights=None),** trọng số này có thể có hoặc không nhưng khi sử dụng trọng số thì độ dài phải bằng với độ dài của data.

**1.2.3 Hàm geometric\_mean(data**)**– hàm tính giá trị trung bình nhân (hình học):**

Hàm statistics.geometric\_mean() là hàm dùng để tính trung bình bằng cách nhân n các số trong tập dữ liệu đã cho và lấy căn thứ n. Về cú pháp sử dụng của hàm này là: **statistics.geometric\_mean(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator). Ngoài ra,** nếu data trống, giá trị âm hoặc 0 thì hàm trả về StatisticsError.

Ví dụ: Cho Gmean1 = [1, 3, 5, 7, 9] tính trung bình nhân (trung bình hình học).

=



Hình 5: Input của hàm trung bình nhân

Text

Description automatically generated

Hình 6: Output của hàm trung bình nhân

**1.2.4 Hàm harmonic\_mean(data**)**– hàm tính giá trị trung bình điều hòa:**

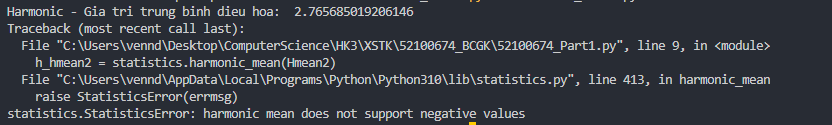
Hàm statistics.harmonic\_mean()là hàm dùng để tính giá trị trung bình điều hòa (vị trí trung tâm) của tập dữ liệu đã cho. Là giá trị nghịch đảo của mean() của các nghịch đảo của một tập hợp dữ liệu quan sát nhất định. Về cú pháp sử dụng của hàm này là: **statistics.harmonic\_mean(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator). Ngoài ra,** nếu data trống và giá trị âm kết quả trả về StatisticsError.

Ví dụ: Cho danh sách 6 số: Hmean1= [1, 2, 3, 9, 10, 8]

Text

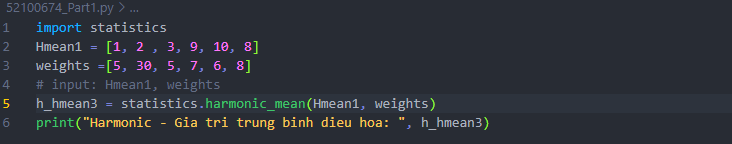
Description automatically generated

Hình 7: Input của hàm harmonic



Hình 8: Output của hàm harmonic

Ngoài ra, đối python version 3.10 còn hỗ trợ hàm có trọng số. Cú pháp sử dụng là **statistics.harmonic\_mean(data,weights=None),** trọng số này có thể có hoặc không nhưng khi sử dụng trọng số thì độ dài phải bằng với độ dài của data.



Hình 9: Input của hàm harmonic có trọng số



Hình 10:Output của hàm harmonic có trọng số

**1.2.5 Hàm median() – hàm tính giá trị trung vị:**

Hàm statistics.meadian() tính giá trị trung vị từ danh sách dữ liệu chưa được sắp xếp. Khi phần tử của dữ liệu là số lẻ, giá trị giữa được trả về. Ngược lại, nó trả về trung bình cộng hai giá trị ở giữa. Cú pháp của hàm là: **statistics.meadian(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator). Ngoài ra,** data trống sẽ xảy ra StatisticsError và kết quả trả về của hàm này là kiểu số thực.

Ví dụ: Cho danh sách chưa sắp xếp Median1 = [1,2,0,9,10,8]

Sắp xếp: [0,1,2,8,9,10] => tính trung vị list có phần tử chẵn (2+8)/2=5.0

Text

Description automatically generated

Hình 11: Input của hàm median

Text

Description automatically generated

Hình 12: Output của hàm median

**1.2.6 Hàm meadian\_low() – hàm tính giá trị trung bình thấp:**

Hàm statistics.meadian\_low() tính giá trị trung vị thấp từ danh sách dữ liệu chưa được sắp xếp. Khi phần tử của dữ liệu là số lẻ, giá trị giữa được trả về. Ngược lại, trả về giá trị thấp hơn giữa hai giá trị ở giữa. Cú pháp của hàm: **statistics.meadian\_low(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và data** trống sẽ xảy ra StatisticsError.

Ví dụ: Cho danh sách chưa sắp xếp Medianlow1 = [1,2,0,9,10,8]

Sắp xếp: [0,1,2,8,9,10] => tính trung vị list có phần tử chẵn nó sẽ về 2

Text

Description automatically generated

Hình 13: Input của hàm median\_low

Text

Description automatically generated

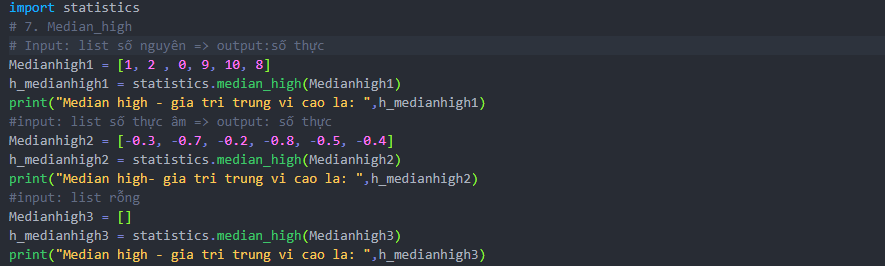
Hình 14: Output của meadian\_low

**1.2.7 Hàm meadian\_high() – hàm tính giá trị trung bình cao:**

Hàm statistics.meadian\_high() tính giá trị trung vị cao từ danh sách dữ liệu chưa được sắp xếp. Khi phần tử của dữ liệu là số lẻ, giá trị giữa được trả về. Ngược lại, nó trả về giá trị cao hơn của hai giá trị ở giữa. Cú pháp của hàm: **statistics.meadian\_high(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và data** trống sẽ xảy ra StatisticsError.

Ví dụ: Cho danh sách chưa sắp xếp Medianhigh1 = [1,2,0,9,10,8]

Sắp xếp: [0,1,2,8,9,10] => tính trung vị list có phần tử chẵn nó sẽ về 8



Hình 15: Input của hàm median\_high

Text

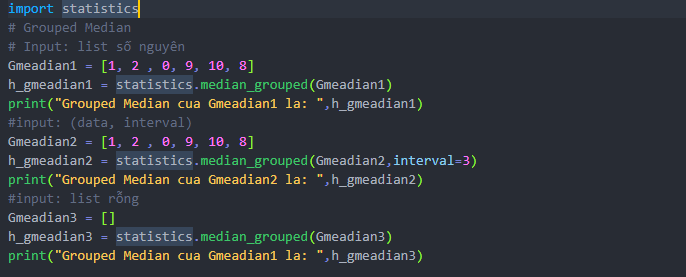
Description automatically generated

Hình 16: Output của hàm median\_high

**1.2.8 Hàm meadian\_grouped():**

Hàm statistics.meadian\_grouped() là hàm trả về giá trị trung bình của dữ liệu liên tục được nhóm lại, được tính bằng phân vị thứ 50. Cú pháp của hàm tính: **statistics.meadian\_grouped (data, interval)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) còn interval là** khoảng đối số tùy chọn đại diện cho khoảng lớp và nếu không thiết lập nó sẽ mặc định là 1. Thay đổi khoảng lớp tự nhiên sẽ thay đổi nội suy và data trống sẽ xảy ra StatisticsError.

Ví dụ: Cho danh sách Gmeadian1 = [1,2,0,9,10,8] và Gmeadian2 = [1,2,0,9,10,8] tính median\_grouped của Gmeadian1(data) và Gmedian3(data,interval=3)



Hình 17: Input của hàm median\_grouped

Text

Description automatically generated

Hình 18: Output của hàm median\_high

**1.2.9 Hàm mode() – hàm trả về giá trị xuất hiện nhiều nhất:**

Hàm statistics.mode() là hàm trả về giá trị xuất hiện nhiều nhất. Cú pháp của hàm tính: **statistics.mode(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator)**. Tham số đầu vào, data trống sẽ xảy ra StatisticsError.

Ví dụ: Cho danh sách mode1 = [1, 5, 6, 7, 6, 2, 1, 2, 3, 2, 5]

Số 2 được lặp đi lặp lại 3 lần => nhiều nhất nên 2 là mode

Text

Description automatically generated

Hình 19: Input của hàm mode

Text

Description automatically generated

Hình 20: Output của hàm mode

**1.2.10 Hàm multimode() – hàm trả về danh sách giá trị xuất hiện thường xuyên:**

Hàm statistics.multimode() là hàm trả về danh sách các giá trị xuất hiện thường xuyên nhất theo thứ tự lần đầu tiên chúng gặp trong dữ liệu. Cú pháp của hàm tính: **statistics.multimode(data)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator)**. Nếu tham số đầu vào rỗng thì hàm cũng trả về list rỗng.

Ví dụ: Cho danh sách multimode1 = ['tranthivensddhchobduocdegapvha']

Lần lượt các kí tự xuất hiện nhiều nhất là: a: 4, h: 4, d: 4 nên 3 kí tự này là mode

Text

Description automatically generated

Hình 21: Input của hàm multimode

Graphical user interface

Description automatically generated with low confidence

Hình 22: Output của hàm multimode

**1.2.11 Hàm quantiles() -  những giá trị (hay điểm cắt (cut points))**:

Hàm statistics.quantiles() là hàm chia dữ liệu thành n khoảng liên tục với xác suất bằng nhau và trả về danh sách các điểm cắt tách các khoảng n-1. Cú pháp của hàm: **statistics.quantiles(data, \*, n=4, method='exclusive')** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator)**. Còn n là số lượng tử, theo mặc định là 4 (tham số này bạn có thể thay đổi), method: phương pháp tính toán các lượng tử được thiết lập mặc định. Nếu như data không đủ 2 phần tử hoặc n nhỏ hơn 1 hàm sẽ trả về StatisticsError.

Ví dụ: Cho quantile1 = [1, 3, 5, 7, 9, 11, 10] và tính danh sách các điểm cắt tách các khoảng n-1.

Text

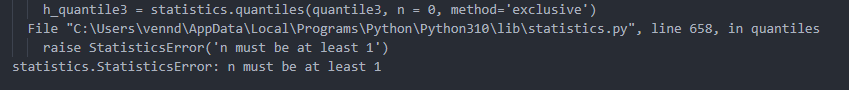
Description automatically generated

Hình 23: Input của hàm quantiles

Text

Description automatically generated

Hình 24:Output của hàm quantiles

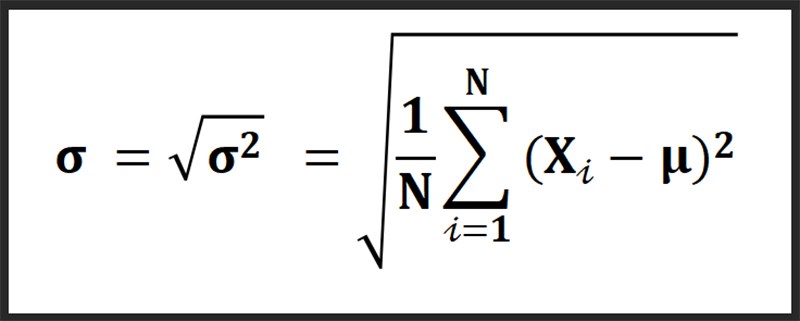


Hình 25: Output của hàm quantiles

**1.2.12 Hàm pstdev() - hàm tính độ lệch chuẩn tổng thể:**

Hàm statistics.pstdev() là hàm trả về độ lệch chuẩn tổng thể - đo mức độ phân tán của một tập dữ liệu quanh giá trị trung bình (Mean) được tính bằng căn bậc hai của phương sai. Cú pháp của hàm là: **statistics.pstdev(data, mu = None)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và mu là** đối số tùy chọn đó là giá trị mà từ đó độ lệch được tính toán nếu bỏ qua sẽ thu được giá trị trung bình của dữ liệu. **Ngoài ra,** nếu data trống nó trả về StatisticsError. Kết quả trả về giá trị thực.

Ví dụ: Cho Pstdev1 = [1, 2, 5, 4, 8, 9, 12]. Tính độ lệch chuẩn tổng thể?



Text

Description automatically generated

Hình 26: Input của hàm pstdev

Text

Description automatically generated

Hình 27: Output của hàm pstdev

**1.2.13 Hàm pvariance() – tính phương sai tổng thể:**

Hàm statistics.pvariance() là hàm trả về phương sai tổng thể thực tế của dữ liệu hay thời điểm thứ hai trả về giá trị trung bình, là thước đo tính biến thiên (trải rộng hoặc phân tán) của dữ liệu. Một phương sai lớn chỉ ra rằng dữ liệu được trải rộng, một phương sai nhỏ cho thấy nó được tập hợp chặt chẽ xung quanh giá trị trung bình. Cú pháp của hàm là: **statistics.pvariance(data, mu = None)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và mu là** đối số tùy chọn đó là giá trị mà từ đó độ lệch được tính toán nếu bỏ qua sẽ thu được giá trị trung bình của dữ liệu. **Ngoài ra,** nếu data trống nó trả về StatisticsError.

Sử dụng hàm này để tính toán phương sai từ toàn bộ tập hợp. Để ước tính phương sai từ một mẫu, variance() hàm thường là một lựa chọn tốt hơn.

Ví dụ: Cho Pvariance1 = [1,3,5,7,8,9,9,0,5,6]. Tính phương sai tổng thể thực tế.

Text

Description automatically generated

Hình 28: Input của hàm Pvariance

Text

Description automatically generated

Hình 29: Output của hàm Pvariance

**1.2.14 Hàm stdev() - hàm tính độ lệch chuẩn mẫu:**

Hàm statistics.stdev() là hàm trả về độ lệch tiêu chuẩn mẫu - nó được sử dụng để định lượng thước đo mức độ lan truyền, sự biến đổi của một tập hợp các giá trị dữ liệu. Cú pháp của hàm là: **statistics.stdev(data, xbar=None)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và xbar là** đối số tùy chọn đó là giá trị mà từ đó độ lệch được tính toán nếu bỏ qua sẽ thu được giá trị trung bình của dữ liệu. **Ngoài ra,** nếu data dưới 2 phần tử nó trả về StatisticsError. Kết quả trả về giá trị thực.

Ví dụ: Cho Stdev1 = [1, 2, 5, 4, 8, 9, 12]. Tính độ lệch chuẩn?

Text

Description automatically generated

Hình 30: Input của hàm stdev

Text

Description automatically generated

Hình 31: Output của hàm stdev

**1.2.15 Hàm variance() – tính phương sai mẫu:**

Hàm statistics.variance() là hàm trả về phương sai mẫu của dữ liệu hay thời điểm thứ hai trả về giá trị trung bình, là thước đo tính biến thiên (trải rộng hoặc phân tán) của dữ liệu. Một phương sai lớn chỉ ra rằng dữ liệu được trải rộng, một phương sai nhỏ cho thấy nó được tập hợp chặt chẽ xung quanh giá trị trung bình. Cú pháp của hàm là: **statistics.variance(data, xbar = None)** trong đó tham số đầu vào data có thể là một tập hợp các số nguyên được tạo thứ tự theo yêu cầu (sequence), danh sách, trình lặp **(iterator) và mu là** đối số tùy chọn đó là giá trị mà từ đó độ lệch được tính toán nếu bỏ qua sẽ thu được giá trị trung bình của dữ liệu. **Ngoài ra,** nếu data ít hơn hai phần tử nó trả về StatisticsError.

Sử dụng hàm này để tính toán phương sai mẫu. Để tính phương sai tổng thì hàm pvariance() hàm thường là một lựa chọn tốt hơn.

Ví dụ: Cho Variance1 = [1,3,5,7,8,9,9,0,5,6]. Tính phương sai mẫu của tập dữ liệu đã cho.

Text

Description automatically generated

Hình 32: Input của hàm Variance

Text

Description automatically generated

Hình 33: Output của hàm Variance

**1.2.16 Hàm covariance() – tính hiệp phương sai:**

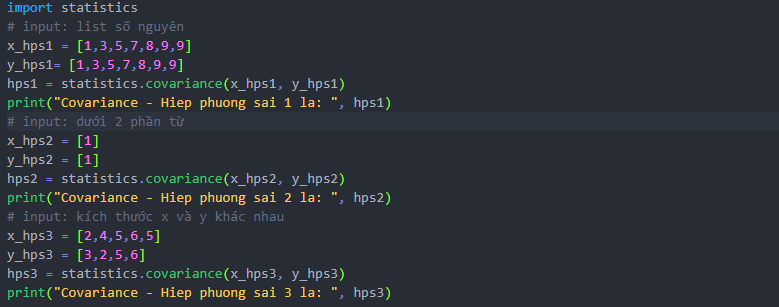
Hàm statistics.covariance() là hàm trả về hiệp phương sai mẫu của hai đầu vào x và y . Hiệp phương sai là thước đo sự thay đổi chung của hai đầu vào.

* Giá trị dương: giá trị hiệp phương sai dương cho thấy rằng cả hai biến ngẫu nhiên đều di chuyển theo cùng một hướng. Nếu một biến ngẫu nhiên tăng thì biến ngẫu nhiên còn lại cũng tăng theo và ngược lại.
* Giá trị âm: giá trị hiệp phương sai âm cho thấy rằng cả hai biến ngẫu nhiên đều di chuyển theo hướng ngược lại. Nếu một biến ngẫu nhiên tăng, thì biến ngẫu nhiên kia giảm. Tương tự, nếu một biến ngẫu nhiên giảm, biến kia tăng.
* Giá trị bằng 0: Khi hai biến ngẫu nhiên độc lập với nhau, hiệp phương sai giữa chúng bằng không.

Cú pháp của hàm là: **statistics.covariance(x,y,/)** trong đó tham số đầu vào hiệp phương sai là thước đo sự thay đổi chung của hai đầu vào. Khi chúng ta sử dụng phương thức covariance (), độ dài của cả hai đầu vào phải giống nhau và không dưới hai phần tử.

Tham số đầu vào x và y lần lượt là tập các danh sách có giá trị số

Ví dụ: Cho x\_hps1 = [1,3,5,7,8,9,9], y\_hps1= [1,3,5,7,8,9,9], x\_hps2 = [1], y\_hps2 = [1], x\_hps3 = [2,4,5,6,5], y\_hps3 = [3,2,5,6]. Tính hiệp phương sai của từng cặp x,y.



Hình 34: Input của hàm covariance

Text

Description automatically generated

Hình 35: Output của hàm covariance

Text

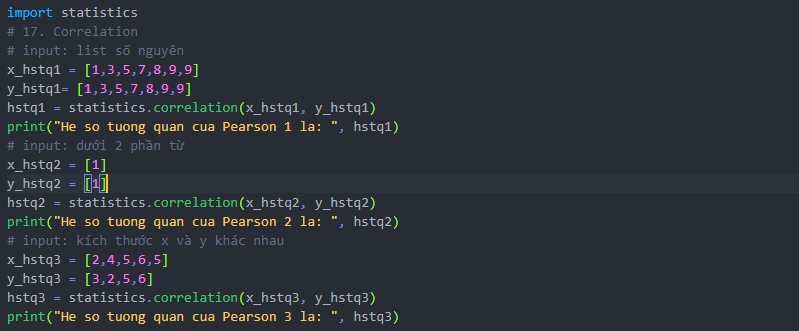
Description automatically generated

Hình 36: Output của hàm covariance

**1.2.17 Hàm correlation() - hàm tính hệ số tương quan của Pearson:**

Hàm statistics.correlation() là hàm trả lại hệ số tương quan của Pearson cho hai đầu vào x và y. Hệ số tương quan r của Pearson nhận các giá trị từ -1 đến +1. Nó đo độ mạnh và hướng của mối quan hệ tuyến tính, trong đó +1 có nghĩa là mối quan hệ tuyến tính rất mạnh, tích cực, -1 rất mạnh, mối quan hệ tuyến tính tiêu cực và 0 không có mối quan hệ tuyến tính. Cú pháp của hàm là: **statistics.correlation(x, y, /)** Khi chúng ta sử dụng phương thức correlation (), độ dài của cả hai đầu vào phải giống nhau và không dưới 2 phần tử. Tham số đầu vào x và y lần lượt là tập các danh sách có giá trị số.

Ví dụ: Cho x\_hstq1 = [1,3,5,7,8,9,9], y\_hstq1= [1,3,5,7,8,9,9], x\_hstq2 = [1] y\_hstq2 = [1], x\_hstq3 = [2,4,5,6,5], y\_hstq3 = [3,2,5,6]. Tính hệ số tương quan của từng cặp x,y.



Hình 37: Input của hàm correlation

Text

Description automatically generated

Hình 38: Output của hàm correlation

Text

Description automatically generated

Hình 39: Output của hàm correlation

**1.2.18 Hàm linear\_regression() – phương trình hồi quy tuyến tính:**

Hàm statistics.linear\_regression() là hàm trả về phương trình tuyến tính dựa trên tập dữ liệu quan hệ giữa x (dữ liệu đầu vào) và y (dữ liệu đầu ra). Cú pháp của hàm: **statistics.linear\_regression(x, y, /, \*, proportional=False)** trong đó x là biến giải thích và y là biến phụ thuộc: y = slope \* x + intercept + noise trong đó slopevà interceptlà các tham số hồi quy được ước tính và noiseđại diện cho sự biến thiên của dữ liệu không được giải thích bởi hồi quy tuyến tính (nó bằng sự khác biệt giữa các giá trị dự đoán và thực tế của biến phụ thuộc). Cả hai đầu vào phải có cùng độ dài (không ít hơn hai) và biến độc lập x không được là hằng số nếu không sẽ StatisticsError. Trong phiên bản 3.11: thêm hỗ trợ cho **proportional.**

Ví dụ: Chúng ta có thể sử dụng ngày dạy của giảng viên để dự đoán số lượng tích lũy các video sẽ được sản xuất vào năm 2022 với giả định rằng chúng đã giữ được tốc độ.

Text

Description automatically generated

Hình 40: Input của hàm hồi quy tuyến tính – linear\_regression

Text

Description automatically generated with medium confidence

Hình 41: Output của hàm hồi quy tuyến tính – linear\_regression

Text

Description automatically generated

Hình 42: Output của hàm hồi quy tuyến tính – linear\_regression

Nếutỷ lệ thuận là đúng, biến độc lập x và biến phụ thuộc y được giả định là tỷ lệ thuận. Dữ liệu vừa với một dòng đi qua điểm gốc. Vì điểm chặn sẽ luôn là 0,0, nên hàm tuyến tính cơ bản đơn giản hóa thành: y = slope \* x + noise

**PHẦN 2: THUẬT TOÁN CÂN BẰNG HISTOGRAM – XỬ LÝ ẢNH**

**2.1 Cơ sở lý thuyết chung về thuật toán cân bằng histogram:**

**2.1.1 Lý thuyết về xử lý ảnh và histogram:**

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người.

Trước khi tìm hiểu về cân bằng histogram, chúng ta cần biết về histogram là gì?

Histogram của ảnh là một cách biểu diễn (bằng hình ảnh) (graphical representation) về phân bố cường độ sáng của ảnh. Trong Histogram của ảnh, trục ngang biểu thị giải giá trị của cường độ sáng (thường là số nguyên từ 0 đến 255) và trục dọc biểu thị cho số pixel có giá trị cường độ sáng ứng với trục ngang.

Ví dụ như hình dưới, có thể thấy:

* Hình ảnh có 4 pixels có giá trị 0, và tương ứng với nó, histogram có một cột có độ cao là 4 tại intensity (cường độ sáng) có giá trị là 0. Tương trị với intensity 50 có 2 pixels, 125 có 1 pixels và 255 có 2 pixels.
* Các giá trị intensity khác không xuất hiện trên ảnh, đồng nghĩa với không có pixels nào có intensity như vậy; do đó, trên histogram, tại các giá trị intesity đó sẽ không có cột giá trị nào.

Chart, bar chart

Description automatically generated

Hình 43: Biểu đồ phân bố pixels

**2.1.2 Lý thuyết về cân bằng histogram:**

**2.1.2.1 Cân bằng histogram:**

**Cân bằng histogram** (histogram equalization) là sự điều chỉnh histogram về trạng thái cân bằng, làm cho phân bố (distribution) giá trị pixel không bị co cụm tại một khoảng hẹp mà được "kéo dãn" ra. Cân bằng histogram là một phương pháp tiền/hậu xử lí ảnh rất mạnh mẽ. Đặc biệt trong nhiều bài toán mình từng làm trong lĩnh vực compute vision, phương pháp tiền xử lí ảnh này cho chất lượng dữ liệu rất cao, cải thiện chất lượng model deep learning rất nhiều.

**2.1.2.2 Ví dụ, nhận xét, phân tích và đánh giá:**

**Ví dụ:**

Diagram

Description automatically generated

Hình 44: Biểu đồ cho thấy độ sáng cũng như histogram của ảnh

**Nhận xét:**

* Với ảnh tối thì histogram có các cột tập trung vào bên trái tương ứng với màu tối, cường độ sáng chỉ tập trung ở vùng tối (0 -> 100).
* Với ảnh sáng thì histogram có tập trung vào bên phải chứa các pixel trắng, cường độ sáng chỉ tập trung ở vùng sáng (120 -> 255).
* Với ảnh độ tương phản thấp (low-contrast) thì histogram có các cột tập trung xít nhau và ở giữa, cường độ sáng chỉ tập trung ở khoảng giữa (80 -> 150)
* Với ảnh độ tương phản cao (high-contrast) thì histogram đều với các giá trị, cường độ sáng phân bố dàn trải từ 0 -> 255.

Ở ví dụ trên, với mắt thường của con người, có thể thấy hình ảnh có độ tương phản cao có chất lượng tốt hơn so với các hình ảnh có độ tương phản thấp. Điều này không chỉ đúng với con người, mà đối với các thuật toán xử lý ảnh, một hình ảnh có độ tương phản cao sẽ giúp các thuật toán xử lý tốt hơn.

**Chú ý và đánh giá**: Histogram equalization chỉ giúp hình ảnh phân tách rõ ràng hơn, nhưng không đảm bảo hình ảnh không bị mất dữ liệu so với hình gốc.

**2.2 Thuật toán Histogram và các bước thực hiện:**

Các bước của thuật toán:

* Bước 1: Gọi hàm cân bằng histogram là F(x) với giá trị x thuộc [0, 255].
* Bước 2: Gọi hàm thống kê pixel cho từng mức sáng là H(x)
* Bước 3: Gọi hàm tích lũy tính từng mức sáng là Z(x) theo công thức:
* Bước 4: Hàm biến đổi F tại một mức sáng x được tính theo công thức:

**F(x) =**

**PHẦN 3: THỰC THI (IMPLEMENTATION)**

**3.1 Giải thích và hướng dẫn chạy code của thuật toán:**

Ta bắt đầu import thư viện và load ảnh (source của ảnh code dưới đây là: anhtest.jpg)

import cv2

import numpy as np

import matplotlib.pyplot as plt

# chèn cv2 để đọc ảnh gốc

histogram\_0 = "anhtest.jpg" # lấy ảnh từ cây thư mục

img1 = cv2.imread(histogram\_0) #ảnh này dùng để in ra so sánh

img = cv2.imread(histogram\_0) #ảnh này dùng để gọi hàm cân bằng

# sau đó tính kích thước của ảnh img1 ban đầu

hist1 = np.zeros((256,), np.uint8)

[width,height] = img1.shape[:2]

# dung vòng lặp duyệt phần tử để tính ra hình ảnh histogram của ảnh đầu

for i in range(width):

    for j in range(height):

        hist1[img1[i][j]] += 1

#  dùng hàm tính kích thước của ảnh cho thuật toán

def getSizeOfImage(img):

    hist = np.zeros((256,), np.uint8)

    [width,height] = img.shape[:2]

# dùng vòng duyệt phần tử để tính ra hình ảnh histogram

    for i in range(width):

        for j in range(height):

            hist[img[i][j]] += 1

    hist = hist.ravel()

# dùng mảng không để lưu trữ histogram

    array = np.zeros\_like(hist, np.float64)

# hàm biến đổi theo bước 4 của lý thuyết (công thức)

    for i in range(len(array)):

        array[i] = hist[:i].sum()

    new\_hist = ((array - array.min()) / (array.max() - array.min())) \* 255

    new\_hist = np.uint8(new\_hist)

    return new\_hist

# Cân bằng historgram/ gọi hàm tính kích thước về

histogram\_1 = getSizeOfImage(img)

[width,height] = img.shape[:2]

#  tính các giá trị của histogram lúc sau

for i in range(width):

   for j in range(height):

        img[i,j] = histogram\_1[img[i][j]]

# Dùng để xuất hiện ảnh để so sánh kết quả

fig = plt.figure()

plt.subplot(221)  #chia bố cực trên kết quả trả về

plt.title("Ảnh trước khi cân bằng")

plt.imshow(img1, cmap = 'gray')

plt.subplot(222)

plt.title("Ảnh sau khi cân bằng")

plt.imshow(img, cmap='gray')

plt.subplot(223)

plt.title("Histogram trước khi cân bằng")

plt.plot(hist1)

plt.subplot(224)

plt.plot(histogram\_1)

plt.title("Histogram sau cân khi bằng")

plt.tight\_layout() # để căn chỉnh layout cho kết quả trả về

plt.show()

**3.2 Kết quả thử nghiệm:**

Graphical user interface, chart

Description automatically generated

Hình 45: Ảnh xám trắng đen bị mờ trước và sau cân bằng

A picture containing chart

Description automatically generated

Hình 46: Ảnh hội nghị mờ được chỉnh rõ hơn