

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN



## BÀI TẬP LỚN

**TÌM HIỂU THƯ VIỆN STATISTICS CỦA PYTHON VÀ  
THUẬT TOÁN CÂN BẰNG HISTOGRAM TRÊN ẢNH XÁM**

**Môn: Thực hành Xác suất và thống kê ứng dụng cho CNTT**

Sinh viên thực hiện: **Tô Vĩnh Khang – MSSV: 51800408**

Lớp: **18050203** - Nhóm: **11** Tố: **01**

Khoá: **22**

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 04/2019**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



# **BÀI TẬP LỚN**

**TÌM HIỂU THƯ VIỆN STATISTICS CỦA PYTHON VÀ  
THUẬT TOÁN CÂN BẰNG HISTOGRAM TRÊN ẢNH XÁM**

**Môn: Thực hành Xác suất và thống kê ứng dụng cho CNTT**

Sinh viên thực hiện: **Tô Vĩnh Khang – MSSV: 51800408**

Lớp: **18050203** - Nhóm: **11** Tố: **01**

Khoá: **22**

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 04/2019**

## LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Huỳnh Quang Đức đã giảng dạy và hướng dẫn em trong suốt thời gian học môn học này. Giúp em có thêm kiến thức về ngôn ngữ lập trình Python. Cuối cùng em xin cảm ơn Nhà trường và Khoa CNTT tạo điều kiện cho em được học tập và hoàn thành tốt môn học Xác suất thống kê Ứng dụng cho CNTT.

## BÀI TẬP LỚN CUỐI KÌ ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Em xin cam đoan đây là sản phẩm của riêng em và được sự hướng dẫn của Thầy Huỳnh Quang Đức. 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 em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài báo cáo này 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 bài tập lớn 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 16 tháng 04 năm 2019

Tác giả

(ký tên và ghi rõ họ tên)

Tô Vĩnh Khang

## 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

Tìm hiểu về các hàm trong thư viện Statistics của Python. Tương ứng mỗi hàm là các giải thích về chức năng, cách sử dụng (tham số đầu vào, ý nghĩa kết quả trả về), code ví dụ minh họa để người đọc có thể hiểu sâu hơn về thư viện trên.

## MỤC LỤC

LỜI CẢM ƠN.....	2
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN.....	3
TÓM TẮT.....	4
MỤC LỤC.....	5
DANH MỤC HÌNH.....	6
CHƯƠNG 1 – CÁC HÀM TRONG THU VIỆN STATISTICS CỦA PYTHON.....	7
1.1 PHƯƠNG PHÁP TÌM VỊ TRÍ TRUNG TÂM.....	
1.1.1 HÀM TÌM GIÁ TRỊ TRUNG BÌNH SỐ HỌC.....	
1.1.1.1 Hàm mean.....	8
1.1.1.2 Hàm harmonic_mean.....	9
1.1.2 HÀM TÌM TRUNG VỊ.....	
1.1.2.1 Hàm median.....	10
1.1.2.2 Hàm median_low.....	11
1.1.2.3 Hàm median_high.....	12
1.1.2.4 Hàm median_grouped.....	12
1.1.2.5 Hàm mode.....	15
1.2 PHƯƠNG PHÁP TÌM PHƯƠNG SAI VÀ ĐỘ LỆCH CHUẨN.....	
1.2.1 HÀM TÌM PHƯƠNG SAI.....	16
1.2.1.1 Hàm pvariance.....	
1.2.1.2 Hàm variance.....	
1.2.2 HÀM TÌM ĐỘ LỆCH CHUẨN.....	18
1.2.2.1 Hàm pstdev.....	
1.2.2.2 Hàm stdev.....	
1.3 EXCEPTION - Hàm StatisticsError.....	20
CHƯƠNG 2 – CÂN BẰNG ẢNH BẰNG PHƯƠNG PHÁP HISTOGRAM.....	23
2.1 PHƯƠNG PHÁP HISTOGRAM VÀ CÁCH SỬ DỤNG.....	
2.2 TÀI LIỆU THAM KHẢO.....	27

## DANH MỤC HÌNH

Hình 1.1 Khai báo thư viện staticitis của python.....	7
Hình 1.2 Mẫu ví dụ về số trung bình cộng.....	8
Hình 1.3 Code minh họa về hàm mean.....	
Hình 1.4 Code minh họa về hàm harmonic_mean .....	9
Hình 1.5 Minh họa về trung vị.....	10
Hình 1.6 Code minh họa về hàm median.....	
Hình 1.7 Code minh họa về hàm median_low.....	11
Hình 1.8 Code minh họa về hàm median_high.....	12
Hình 1.9 Code minh họa về hàm median_group (phần 1.....	13
Hình 1.10 Code minh họa về hàm median_group (phần 2).....	
<u>Hình 1.11</u> Code minh họa về hàm median_group (phần 3).....	14
<u>Hình 1.12</u> Code minh họa về hàm mode.....	15
<u>Hình 1.13</u> Ví dụ về điểm thi giữa kì của các nhóm sinh viên TDTU.....	
<u>Hình 1.14</u> Code minh họa về hàm pvariance.....	17
<u>Hình 1.15</u> Code minh họa về hàm variance.....	18
<u>Hình 1.16</u> Code minh họa về hàm pstdev.....	19
<u>Hình 1.17</u> Code minh họa về hàm stdev.....	
<u>Hình 1.18</u> Code minh họa về hàm StatisticsError (Exp01).....	20
<u>Hình 1.19</u> Kết quả trả về của hàm StatisticsError (Exp01).....	
<u>Hình 1.20</u> Code minh họa về hàm StatisticsError (Exp02).....	21
<u>Hình 1.21</u> Kết quả trả về của hàm StatisticsError (Exp02).....	
<u>Hình 1.22</u> Code minh họa về hàm StatisticsError (Exp03).....	22
<u>Hình 1.23</u> Kết quả trả về của hàm StatisticsError (Exp03).....	
<u>Hình 2.1</u> Công thức mô tả tỷ lệ số pixel mức xám với tổng pixel ảnh.....	23
<u>Hình 2.2</u> Một vài hình dạng Histogram.....	24
<u>Hình 2.3</u> Ảnh minh họa trước và sau khi điều chỉnh độ tương phản.....	25
<u>Hình 2.4</u> Một mẫu code minh họa cho phương pháp cân bằng ảnh Histogram.....	26

## CHƯƠNG 1: CÁC HÀM TRONG THƯ VIỆN STATISTICS CỦA PYTHON

Statistics (thống kê) là một nhánh của toán học, thực hiện thu thập, phân tích, thấu hiểu, và biểu diễn một lượng số liệu lớn. Và thư viện statistics trong python sẽ hỗ trợ chúng ta thực hiện điều đó.

Để sử dụng được chúng , trước tiên ta cần khai báo thư viện statistics của python như sau:

```
import statistics
```

Hình 1.1 Khai báo thư viện statistics của python

### **1.1 PHƯƠNG PHÁP TÌM VỊ TRÍ TRUNG TÂM**

Trong toán học , có khá nhiều cách để tìm vị trí trung tâm của một tập hợp n số. Thông qua thư viện Statistics của Python , ta có thể tìm hiểu qua hai dạng phổ biến là giá trị trung bình và trung vị số học.

#### ***1.1.1.. CÁC HÀM TÌM GIÁ TRỊ TRUNG BÌNH SỐ HỌC***

Trung bình số học là tổng giá trị của các phần tử trong tập dữ liệu chia cho số phần tử của tập dữ liệu đó. Tuy rằng nó chỉ là 1 thuật toán tính trung bình trong số nhiều thuật toán tính trung bình khác nhưng nó có thể tính toán tìm ra được vị trí trung tâm của tập dữ liệu.

### 1.1.1.1 Hàm mean

Số trung bình cộng của một dãy số trong toán học là tỉ số giữa tổng giá trị của tập hợp số đó và toàn bộ các phần tử có trong tập hợp đó. Nói cách khác, số trung bình cộng chính là thương giữa tổng các số hạng có trong dãy số đã cho với số các số hạng vừa lấy tổng.

$$\frac{M + E + A + N}{4}$$

Hình 1.2 Mẫu ví dụ về số trung bình cộng

Tên hàm: ***statistics.mean(data)***

Chức năng: Tìm giá trị trung bình cộng của một dãy số.

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

```
print("Hàm mean: ")
print(statistics.mean([0,1,2,3,4,5,6,7,8,9])) # Tham số đầu vào là 1 dãy số 0 -> 9.
print(statistics.mean(i for i in range(10))) # Tham số đầu vào là 1 vòng lặp từ 0 -> 9.
```

Hình 1.3 Code minh họa về hàm mean

Theo code minh họa trên thì hàm mean này được tính toán như sau:

$$Ans = (0+1+2+3+4+5+6+7+8+9)/10$$

Giá trị trung bình lúc này là ***Ans***, giá trị này cũng chính là tiền đề để ta có thể tính toán nhiều số liệu khác. Đặc biệt là phương sai mẫu số học (variance).

### 1.1.1.2 Hàm harmonic\_mean

Trung bình điều hòa hay còn được gọi là trung bình nghịch đảo. Nó là sự nghịch đảo của trung bình số học của kiểu dữ liệu nghịch đảo. Trung bình điều hòa là một phương pháp tính toán giá trị trung tâm của tập dữ liệu. Thích hợp khi giá trị tính toán là tỉ lệ hoặc tỉ số.

Tên hàm: ***statistics.harmonic\_mean(data)***

Chức năng: Tìm giá trị trung bình điều hòa của một dãy số.

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp của các số thực khác 0. (Xem code minh họa bên dưới)

```
print(" Hàm harmonic_mean: ")

print(statistics.harmonic_mean([1,2,3,4,5,6]))

print(statistics.harmonic_mean(i for i in range(1,7)))
```

Hình 1.4 Code minh họa về hàm harmonic\_mean

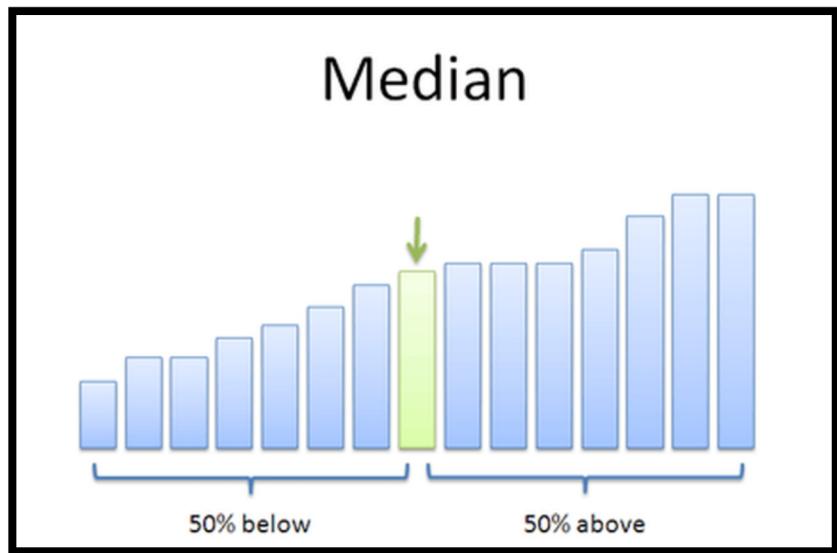
Theo code minh họa trên thì hàm harmonic\_mean này được tính toán như sau:

$$Ans = 6/((1/1)+(1/2)+(1/3)+(1/4)+(1/5)+(1/6))$$

### 1.1.2.. CÁC HÀM TÌM TRUNG VỊ

Trong lý thuyết xác suất và thống kê, số trung vị là một số tách giữa nửa lớn hơn và nửa bé hơn của một mẫu, một quần thể, hay một phân bố xác suất. Nó là giá trị giữa trong một phân bố, mà số các số nằm trên hay dưới con số đó là bằng nhau. Điều đó có nghĩa rằng 1/2 quần thể sẽ có các giá trị nhỏ hơn hay bằng số trung vị, và một nửa quần thể sẽ có giá trị bằng hoặc lớn hơn số trung vị.[2]

### 1.1.2.1 Hàm median



Hình 1.5 Minh họa về trung vị  
(Nguồn: IB Geography)

Tên hàm: ***statistics.median(data)***

Chức năng: Tìm giá trị trung vị của 1 dãy số bằng phương pháp "Trung bình cộng của 2 phần tử nằm giữa".

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

```
print(" Hàm median: ")

print(statistics.median([1,4,7,9]))

print(statistics.median(i for i in range(1,6)))
```

bUTUE(RERTSCTCSTWEDTSU(T LOU T TU LGSU8E(T^P)))

Hình 1.6 Code minh họa về hàm median

Theo code minh họa thì:

Dãy số 1,4,7,9 (4 phần tử - chẵn) thì nó sẽ lấy trung bình cộng của 4 và 7 (*Ans1*).

Và dãy số 1,2,3,4,5 (5 phần tử - lẻ) thì nó sẽ lấy phần tử nằm giữa dãy số đó (*Ans2*).

$$Ans1 = (4+7)/2 ; Ans2 = 3$$

### 1.1.2.2 Hàm median\_low

Tên hàm: *statistics.median\_low(data)*

Chức năng: Tìm giá trị trung vị thấp.

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

```
print(" Hàm median_low: ")
print(statistics.median_low([2,3,4,7,8,9]))
print(statistics.median_low(i for i in range(1,8)))
```

bạn thử (sử dụng module statistics) *median\_low* với dãy số [2,3,4,7,8,9] và [1,2,3,4,5,6,7]

Hình 1.7 Code minh họa về hàm median\_low

Theo code minh họa thì:

Dãy số 2,3,4,7,8,9 (6 phần tử - chẵn) thì nó sẽ lấy phần tử nhỏ nhất trong 2 phần tử nằm ở giữa của dãy số đó (*Ans1*). Dãy số 1,2,3,4,5,6,7 (7 phần tử - lẻ) thì nó sẽ lấy phần tử nằm giữa dãy số đó (*Ans2*).

$$Ans1 = 4; Ans2 = 4$$

### 1.1.2.3 Hàm median\_high

Tên hàm: ***statistics.median\_high(data)***

Chức năng: Tìm giá trị trung vị cao.

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

```
print("Hàm median_high: ")
print(statistics.median_high([2,3,12,25,78,99]))
print(statistics.median_high(i for i in range(3,11,2)))
```

bạn thử (statistics.median\_high([ 2, 3, 12, 25, 78, 99]))

Hình 1.8 Code minh họa về hàm median\_high

Theo code minh họa thì:

Dãy số 2,3,12,25,78,99 (6 phần tử - chẵn) thì nó sẽ lấy phần tử lớn nhất trong 2 phần tử ở giữa của dãy số đó (*Ans1*). Và dãy số 3,5,7,9,11 (5 phần tử - lẻ) thì nó sẽ lấy phần tử nằm giữa dãy số đó (*Ans2*).

$$Ans1 = 25; \quad Ans2 = 7$$

### 1.1.2.4 Hàm median\_grouped

Tên hàm: ***statistics.median\_grouped(data,interval=1)***

Chức năng: Tìm giá trị trung vị của dữ liệu liên tục được nhóm, được tính là phân vị thứ 50, sử dụng phép nội suy.

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

*Nội suy là phương pháp ước tính giá trị của các điểm dữ liệu chưa biết trong phạm vi của một tập hợp rời rạc chia một số điểm dữ liệu đã biết.[3]*

Thuật toán được sử dụng trong hàm median\_group này là:

$$Me = LMe + hMe \times \frac{\frac{n}{2} - Cf b}{fMe}$$

```
print(statistics.median_grouped([5,7,9,11,12,13,14],interval=0))
print(statistics.median_grouped([5,7,9,11,12,13,14],interval=1))
print(statistics.median_grouped([5,7,9,11,12,13,14],interval=2))
print(statistics.median_grouped([5,7,9,11,12,13,14],interval=9999))
```

Hình 1.9 Code minh họa về hàm median\_group (phần 1)

Tham số truyền vào là các giá trị **khác nhau** và tổng số lượng phần tử trong dãy số là **số lẻ** thì kết quả trả về sẽ là giá trị ở giữa dãy số đó và không quan trọng interval truyền vào là bao nhiêu vì nó sẽ không làm thay đổi phép nội suy của bài toán.

Theo code minh họa trên thì:

Dãy số 5,7,9,11,12,13,14 (7 phần tử - lẻ) và interval bất kì đều cho ra kết quả sau:

***Ans = 11.0***

\*Với **interval = 0** thì kết quả trả về sẽ là median, median high hoặc median low tùy vào tổng số lượng phần tử truyền vào.

```
print(statistics.median_grouped([2,3,5,7,9,11,12,13,14,15],interval=0))
print(statistics.median_grouped([2,3,5,7,9,11,12,13,14,15],interval=1))
print(statistics.median_grouped([2,3,5,7,9,11,12,13,14,15],interval=2))
print(statistics.median_grouped([2,3,5,7,9,11,12,13,14,15],interval=9999))
```

Hình 1.10 Code minh họa về hàm median\_group (phần 2)

Tham số truyền vào là các giá trị **khác nhau** và tổng số lượng phần tử trong dãy số là **số chẵn** thì kết quả trả về sẽ là giá trị trung vị nhóm và interval lúc này sẽ làm thay đổi phép nội suy của bài toán.

Theo code minh họa trên thì:

Dãy số 2,3,5,7,9,11,12,13,14,15 (10 phần tử - chẵn) và interval bất kì sẽ cho ra kết quả khác nhau lần lượt như sau:

$$Ans1 = 11.0 ; Ans2 = 10.5$$

$$Ans3 = 10.0 ; Ans4 = -4988.5$$

```
print(statistics.median_grouped([5,7,9,11,11,13,14],interval=0))
print(statistics.median_grouped([5,7,9,11,11,13,14],interval=1))
print(statistics.median_grouped([5,7,9,11,11,13,14],interval=2))
print(statistics.median_grouped([5,7,9,11,11,13,14],interval=9999))
```

Hình 1.11 Code minh họa về hàm median\_group (phần 3)

Tham số truyền vào có từ một giá trị **giống nhau** trở lên thì tổng số lượng phần tử trong dãy số là **số lẻ** hoặc **số chẵn** đều sẽ cho ra kết quả trả về khác nhau và phụ thuộc vào interval truyền vào là bao nhiêu.

Theo code minh họa trên thì:

Dãy số 5,7,9,11,11,13,14 cùng với interval bất kì sẽ cho ra kết quả khác nhau lần lượt như sau:

$$Ans1 = 11.0 ; Ans2 = 10.75$$

$$Ans3 = 10.5 ; Ans4 = -2488.75$$

### 1.1.2.5 Hàm mode

Tên hàm: ***statistics.mode(data)***

Chức năng: Tìm giá trị xuất hiện thường xuyên nhất trong tập hợp, danh sách các giá trị, phần tử. Trong trường hợp không có giá trị nào được lặp lại thì không có mode.

Cách sử dụng: Tham số đầu vào là một chuỗi dữ liệu. (Xem code minh họa bên dưới)

```
print(" Hàm mode: ")

print(statistics.mode([2,5,44,9,11,9,9,5,7]))

print(statistics.mode(["red","blue","green","red","white"]))
```

Hình 1.12 Code minh họa về hàm mode

Theo code minh họa trên thì kết quả trả về là những giá trị được lặp lại nhiều lần nhất , kết quả lần lượt như sau:

$$Ans1 = 9 ; Ans2 = red$$

#### SINH VIÊN NHÓM 1:

10 10 10 10 10

- Trung bình: 10
- Trung vị: 10
- Mode: None

#### SINH VIÊN NHÓM 2:

9 9 9 9 10

- Trung bình: 9.2
- Trung vị: 9
- Mode: 9

#### SINH VIÊN NHÓM 3:

9 10 10 10 10

- Trung bình: 9.8
- Trung vị: 10
- Mode: 10

#### SINH VIÊN NHÓM 4:

8 8 9 10 10

- Trung bình: 9
- Trung vị: 9
- Mode: 8 và 10

Hình 1.13 Ví dụ về điểm thi giữa kì của các nhóm sinh viên TDTU

## 1.2 PHƯƠNG PHÁP TÌM PHƯƠNG SAI VÀ ĐỘ LỆCH CHUẨN

*Trong lý thuyết xác suất và thống kê, phương sai của một biến ngẫu nhiên là một độ đo sự phân tán thống kê của biến đó, nó hàm ý các giá trị của biến đó thường ở cách giá trị kỳ vọng bao xa. Phương sai của một biến ngẫu nhiên là bình phương của độ lệch chuẩn.*

*Độ lệch chuẩn là một đại lượng thống kê mô tả dùng để đo mức độ phân tán của một tập dữ liệu đã được lập thành bảng tần số. Có thể tính ra độ lệch chuẩn bằng cách lấy căn bậc hai của phương sai. Độ lệch chuẩn còn được sử dụng khi tính sai số chuẩn. Khi lấy độ lệch chuẩn chia cho căn bậc hai của số lượng quan sát trong tập dữ liệu, sẽ có giá trị của sai số chuẩn.[4]*

### 1.2.1 HÀM TÌM PHƯƠNG SAI

#### 1.2.1.1 Hàm pvariance

Tên hàm: **statistics.pvariance(data, mu=None)**

Chức năng: Tính phương sai từ toàn bộ dân số của dữ liệu. Để ước tính phương sai từ một mẫu

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

*Một phương sai lớn cho thấy rằng dữ liệu được trải rộng ra. Một phương sai nhỏ cho thấy rằng dữ liệu được nhóm chặt chẽ xung quanh giá trị trung bình.*

*Nếu đối số thứ hai tùy chọn **mu** được đưa ra, nó sẽ là giá trị trung bình của dữ liệu.*

*Nếu không có (None) thì giá trị trung bình sẽ tự động được tính.[1]*

```

print(" Hàm pvariance: ")

print(statistics.pvariance([0,2,4,6,5,7,9]))

print(statistics.pvariance(i for i in range(0,10)))

```

bạn tự (statistics.pvariance(tous les tu l'enrange(0,10)))

Hình 1.14 Code minh họa về hàm pvariance

Theo code minh họa thì:

Dãy số 0,2,4,6,5,7,9 sẽ có kết quả là *Ans1*. Và kết quả vòng lặp từ 0 đến 9 là *Ans2*.

$$Ans1 = \sim 7.92 ; Ans2 = 8.25$$

**Lưu ý:** Không sử dụng đối số thứ hai **mu** tùy tiện vì nó có thể cho ra kết quả không đúng hoặc không thể có.

### 1.2.1.2 Hàm variance

Tên hàm: **statistics.variance(data, xbar=None)**

Chức năng: Tính phương sai từ toàn bộ dân số của dữ liệu. Để ước tính phương sai từ một mẫu

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

*Trả về phương sai mẫu của dữ liệu, có thể lặp lại ít nhất hai số có giá trị thực. Phương sai, hoặc khoảng khắc thứ hai về giá trị trung bình, là thước đo độ biến thiên (làn truyền hoặc phân tán) của dữ liệu. Một phương sai lớn chỉ ra rằng dữ liệu được trải ra; một phương sai nhỏ cho thấy nó được nhóm chặt chẽ xung quanh giá trị trung bình.[1]*

Nếu `xbar` đối số thứ hai tùy chọn được đưa ra, nó sẽ là giá trị trung bình của dữ liệu. Nếu không có thì giá trị trung bình sẽ tự động được tính. Sử dụng chức năng này khi dữ liệu của bạn là một mẫu từ dân số. Hàm variance này thường là lựa chọn tốt hơn cả.

```
print("Hàm variance: ")

print(statistics.variance([0,2,4,6,5,7,9]))

print(statistics.variance(i for i in range(0,10)))
```

Hình 1.15 Code minh họa về hàm variance

Theo code minh họa thì:

Dãy số 0,2,4,6,5,7,9 sẽ có kết quả là *Ans1*. Và kết quả vòng lặp từ 0 đến 9 là *Ans2*.

$$Ans1 = \sim 9.24 ; Ans2 = \sim 9.17$$

**Lưu ý:** Không sử dụng đối số thứ hai `xbar` tùy tiện vì nó có thể cho ra kết quả không đúng hoặc không thể có (tương tự hàm pvariance phía trên).

## 1.2.2 HÀM TÌM ĐỘ LỆCH CHUẨN

### 1.2.1.3 Hàm pstdev

Tên hàm: `statistics.pstdev(data, mu=None)`

Chức năng: Tính độ lệch chuẩn dân số (căn bậc hai của phương sai dân số).

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp.

Xem hàm pvariance phía trên để biết các đối số và các chi tiết khác.

```

print(" Hàm pstdev: ")

print(statistics.pstdev([0,3,5,7,8,10]))

print(statistics.pstdev(i for i in range(0,10)))

```

Hình 1.16 Code minh họa về hàm pstdev

Theo code minh họa thì:

Dãy số 0,3,5,7,8,10 sẽ có kết quả là *Ans1*. Và kết quả vòng lặp từ 0 đến 9 là *Ans2*.

$$Ans1 = \sim 3.304 ; Ans2 = \sim 2.872$$

### 1.2.1.3 Hàm stdev

Tên hàm: ***statistics.stdev(data, xbar=None)***

Chức năng: Trả về độ lệch chuẩn mẫu (căn bậc hai của phương sai mẫu).

Cách sử dụng: Tham số đầu vào là một dãy số hoặc một vòng lặp. (Xem code minh họa bên dưới)

Xem hàm variance phía trên để biết các đối số và các chi tiết khác.

```

print(" Hàm stdev: ")

print(statistics.stdev([0,3,5,7,8,10]))

print(statistics.stdev(i for i in range(0,10)))

```

Hình 1.17 Code minh họa về hàm stdev

Theo code minh họa thì:

Dãy số 0,3,5,7,8,10 sẽ có kết quả là *Ans1*. Và kết quả vòng lặp từ 0 đến 9 là *Ans2*.

$$\text{Ans1} = \sim 3.619 ; \quad \text{Ans2} = \sim 3.027$$

## 1.2 EXCEPTION - Hàm StatisticsError

Thông thường hàm này sẽ có mặc định trong khi ta gọi các hàm phía trên bị sai tham số đầu vào. Tuy nhiên, nếu ta dùng nó vào mục đích kiểm tra lại ngoại lệ khác thì đây là lúc cần sử dụng hàm này.

Tên hàm: ***statistics.StatisticsError***

Chức năng: Kiểm tra những những ngoại lệ khác ta khi truyền tham số đầu vào không đúng hoặc chưa đúng với chức năng của hàm khác.

Cách sử dụng: Tham khảo hình bên dưới

```
#EXP 01:
list_test1=[1,3,5,7,9]
def TestStatisticsError1(list_test1):
    try:
        print("Result of Mode: ", statistics.mode(list_test1))
    except statistics.StatisticsError as ModeError:
        print("ERROR! : ", ModeError)

TestStatisticsError1(list_test1);
```

TestStatisticsError1([1,3,5,7,9])?

Hình 1.18 Code minh họa về hàm StatisticsError (Exp01)

Như ta được biết thì hàm mode sẽ trả về kết quả là giá trị xuất hiện nhiều nhất. Tuy nhiên ở đoạn code này không có sự trùng lặp các giá trị nên kết quả được hàm StatisticsError này sẽ được trả về như sau:

```
ERROR! : no unique mode; found 5 equally common values
```

Hình 1.19 Kết quả trả về của hàm StatisticsError (Exp01)

Như vậy chúng ta sẽ biết được kết quả trả về nếu hàm mode có tham số truyền vào là những giá trị trùng lặp thông qua đoạn code tiếp theo:

```
#EXP 02:
list_test2=[1,7,7,7,9]
def TestStatisticsError2(list_test2):
    try:
        print("Result of Mode: ", statistics.mode(list_test2))
    except statistics.StatisticsError as ModeError:
        print("ERROR! : ", ModeError)

TestStatisticsError2(list_test2);
TestStatisticsError2([1,7,7,7,9])?
```

Hình 1.20 Code minh họa về hàm StatisticsError (Exp02)

Kết quả của Exp02 là:

```
Result of Mode: 7
```

Hình 1.21 Kết quả trả về của hàm StatisticsError (Exp02)

Với hàm mode này , chúng ta có thể thử nghiệm thêm một ví dụ đó là chuỗi kí tự thông qua đoạn code này:

```
#EXP 03:  
list_words=["pink", "green", "black", "white", "pink", "pink", "brown"]  
def TestStatisticsError3(List_words):  
    try:  
        print("Result of Mode: ", statistics.mode(list_words))  
    except statistics.StatisticsError as ModeError:  
        print("ERROR! : ", ModeError)  
  
TestStatisticsError3(list_words);  
TestStatisticsError3(list_words)?
```

Hình 1.22 Code minh họa về hàm StatisticsError (Exp03)

Vậy kết quả của Exp03 này chắc chắn sẽ là:

Result of Mode: pink

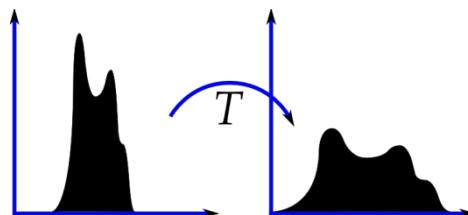
Hình 1.23 Kết quả trả về của hàm StatisticsError (Exp03)

## CHƯƠNG 2 – CÂN BẰNG ẢNH BẰNG PHƯƠNG PHÁP 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. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống. Chẳng hạn như việc cân bằng ảnh bằng phương pháp Histogram để cho ra đời những bức ảnh mới đẹp mắt hơn thông qua việc chỉnh sửa độ tương phản của chúng.[5]*

### 2.1 PHƯƠNG PHÁP HISTOGRAM VÀ CÁCH SỬ DỤNG

*Phương pháp này làm tăng cường độ tương phản toàn cục của bức ảnh, nó có ý nghĩa đặc biệt khi dữ liệu đầu vào có giá trị điểm ảnh co cụm (độ tương phản thấp) ta có thể thấy rõ thông qua biểu đồ histogram. Mục đích của phương pháp cân bằng ảnh Histogram là giúp biến đổi bức ảnh có độ tương phản thấp thành bức ảnh có độ tương phản cao hơn bằng cách trải đều giá trị của pixel, thay vì co cụm mà vẫn giữ được nội dung của bức ảnh.[6]*



Histogram của ảnh xám có các giá trị xám  $r_0, r_1, r_2, \dots, r_{m-1}$  là hàm rời rạc.

$$p(r_k) = \frac{n_k}{n}$$

Hình 2.1 Công thức mô tả tỷ lệ số pixel mức xám với tổng pixel ảnh

$n_k$ : Số lượng pixel ở mức xám  $r_k$

$n$  : Tổng số pixel trong ảnh

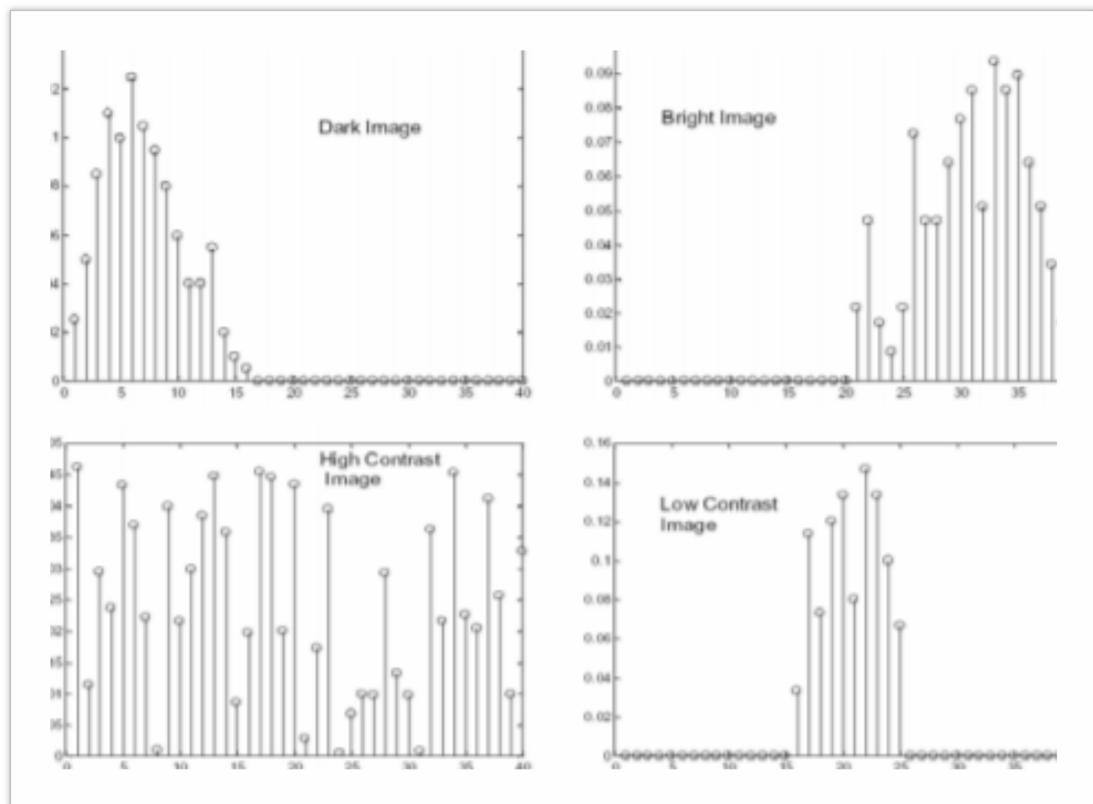
$m$ : Tổng số mức xám

$$S_k = T(r_k) = \sum_{i=0}^k (ni/n) = \sum_{i=0}^k P_{in}(ri) \quad (0 \leq k \leq m - 1)$$

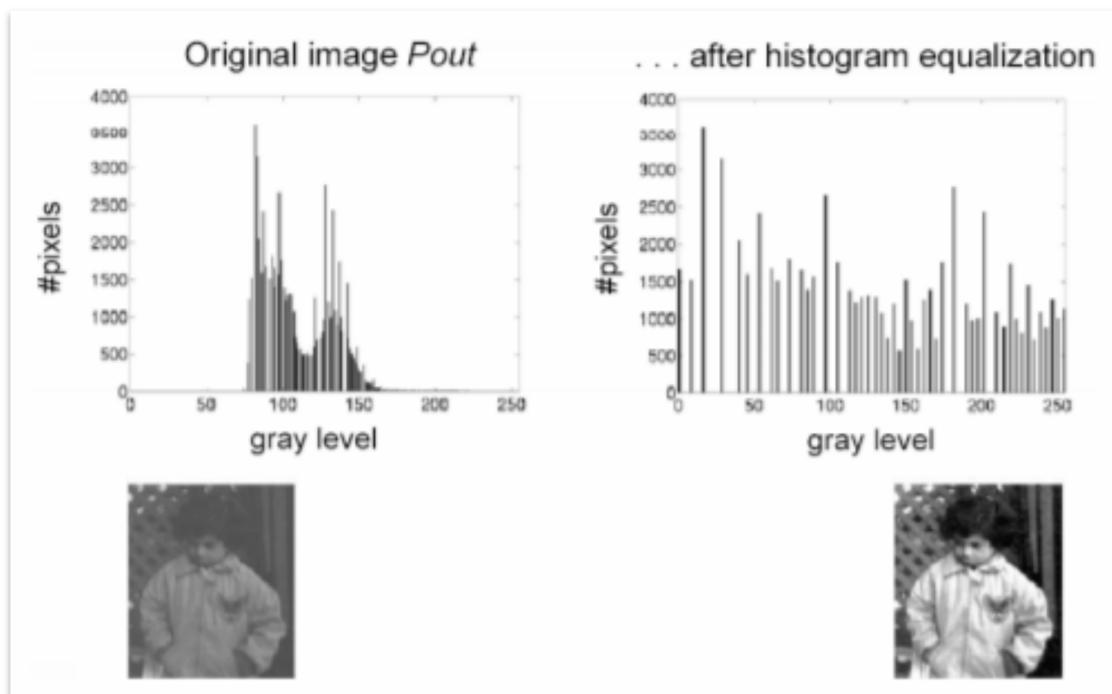
Nếu ta xem mức xám trong ảnh là biến ngẫu nhiên  $R_n$ , mật độ xác suất thì histogram cung cấp như một xấp xỉ của mật độ xác suất.

$$P[R_n=r_k] \sim p(r_k)$$

Phương pháp cân bằng ảnh Histogram này dung ý tưởng biến đổi phi tuyến  $g=T(f)$  áp dụng cho mỗi pixel ảnh  $f(x,y)$  để  $g(x,y)$  phân phối đều.



Hình 2.2 Một vài hình dạng Histogram  
(Nguồn: Khoa Toán Tin – Trường ĐHKHTN 2008)



Hình 2.3 Ảnh minh họa trước và sau khi điều chỉnh độ tương phản  
(Nguồn: Khoa Toán Tin – Trường ĐHKHTN 2008)



Anh ban đầu



Sau khi cân bằng histogram

Có rất nhiều cách để chỉnh sửa ảnh bằng Histogram , nhưng phương pháp phổ biến nhất hiện nay vẫn là tăng cường ảnh bằng cách dùng Histogram cục bộ. Ta có thể dùng hàm mean và hàm variance để xác định độ tương phản cục bộ của một pixel.

Hướng dẫn chạy code:

B1: Ta cần cài đặt thư viện cv2 từ OpenCV trong cmd thông qua lệnh:

***pip install opencv-python (Chỉ modul chính)***

***pip install opencv-contrib-python (Cả modul chính và modul phụ) [7]***

B2: Đưa file hình vào cùng với file code python.

B3: Chính sửa tên file hình sao cho trùng với tên file hình đã được ghi trong source code python hoặc ngược lại.

B4: Ta mở trình biên dịch cmd lên nhập lệnh: ***python <namefilepython>.py***

```

import cv2
import numpy as np
from matplotlib import pyplot as plt

image = cv2.imread('ModelHH.jpg',0);                                #Input to Read your image.
hist,bins = np.histogram(image.flatten(),500,[0,500]);           #CDF (Cumulative Distribution Function).
CDF = hist.cumsum();                                              #cumsum() : Sum of pixel elements
CDF_normal = CDF * hist.max()/ CDF.max();                          #Formula

plt.plot(CDF_normal, color = 'g')                                    #Change color CDF line (green).
plt.hist(image.flatten(),500,[0,500], color = 'y');                #Change color line (yellow).
plt.xlim([0,300])                                                 #Make x axis = 300.
plt.legend(('CDF','HISTOGRAM'), loc = 'center right');
plt.show();

equal = cv2.equalizeHist(image);
SideBySide = np.hstack((image,equal));                               #Stack sidebyside 2 images
cv2.imwrite('ModelHH.png',SideBySide);                                #Output your image.

```

Hình 2.4 Một mẫu code minh họa cho phương pháp cân bằng ảnh Histogram

Trong trường hợp như hình ví dụ minh họa trên thì tên hình sẽ đổi tên thành “ModelHH.jpg”

## 2.2 TÀI LIỆU THAM KHẢO

- [1] <https://docs.python.org/3/library/statistics.html>
- [2] [https://vi.wikipedia.org/wiki/S%C3%A1n\\_trung\\_v%C3%AC\\_E1%BB%8B](https://vi.wikipedia.org/wiki/S%C3%A1n_trung_v%C3%AC_E1%BB%8B)
- [3] [https://vi.wikipedia.org/wiki/Nh%E1%BB%99i\\_suy](https://vi.wikipedia.org/wiki/Nh%E1%BB%99i_suy)
- [4]  
[https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99\\_l%E1%BB%87ch\\_chu%E1%BA%A9n](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_l%E1%BB%87ch_chu%E1%BA%A9n)
- [5] [https://vi.wikipedia.org/wiki/X%C3%A1n\\_BB%AD\\_l%C3%A0n\\_BB%BD](https://vi.wikipedia.org/wiki/X%C3%A1n_BB%AD_l%C3%A0n_BB%BD)
- [6] <http://www.math.hcmuns.edu.vn/~ptbao/BHXLA/3.pdf>
- [7] <https://pypi.org/project/opencv-python/>