VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**DISCRETE STRUCTURES ESSAY**

*Supervisor:* **MR.TRAN HONG TAI**

*Author*: **NGUYEN MINH NHUT– 518H0545**

Class**: 18H50203**

Course **:22**

**HO CHI MINH CITY, 2019**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**DISCRETE STRUCTURES ESSAY**

*Supervisor*:  **MR.TRAN HONG TAI**

*Author*: **NGUYEN MINH NHUT**

Class**: 18H50203**

Course**:22**

**HO CHI MINH CITY, 2019**

Appreciation Letter

Firstly, this should be an honor to send my regards to the Faculty of Information Technology, lecturers and staff from all departments of Ton Duc Thang University. I would like to express my sincere thanks for the support and assistance during the implementation of the statistics and probability report.

I would like to express my gratitude to Mr. Tran Hong Tai - teachers who directly instructed and supervised me to complete this essay.

I sincerely thank my friends and classmates who are studying and working at Ton Duc Thang University and the family has encourage, facilitated and helped me during the process.

Due to the fact that my actual ability is still weak, I ensure that I still have many shortcomings, so I hope my supervisor and the other professors will ignore it. At the same time, I hope to receive many comments from many sources to help me accumulate more experience to complete the upcoming graduation report to achieve better results.

**THE ESSAY HAS BEEN CONDUCTED IN TON DUC THANG UNIVERSITY**

I assure that this is my own product and has been guided by Mr. Tran Hong Tai. The research contents, results in this topic are all about honesty and this is the first time my essay published as an academic essay. The data in the tables for analysis, comments and evaluation are collected by the me from various sources in the reference section.

In addition, comments and assessments as well as data from other authors or organizations are also used in the essay but with references and annotations.

**If there is any fraud is detected, I ensure my complete responsibility for the contents of my work.** Ton Duc Thang University is not related to violations of authority and copyright caused by me during my work process (if any).

*Ho Chi Minh city, Tuesday, 7th October. 2019*

*Author*

*(Sign and provide full name)*



*Nguyen Minh Nhut*

VERIFICATION AND EVALUATION FROM LECTURER

**Supervisor’s evaluation**

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

Ho Chi Minh city, date:

(Sign and provide full name)

**Marking lecturer’s evaluation**

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

Ho Chi Minh city, date:

(Sign and provide full name)

SUMMARY

In this essay, I am going to satisfy these two requirements below:

Requirement 1: Survey all functions in the Statistics library of python.

Correspondingly, each function needs to be explained about its usage, input

parameters, meaning of returned results, and example code.

Requirement 2: Survey about Histogram equalization algorithm for image

processing.

- Survey about the problem, Histogram approach, formula, and algorithm.

- Write an illustrative program.

TABLE OF CONTENTS

[Appreciation Letter i](#_Toc6592836)

[VERIFICATION AND EVALUATION FROM LECTURER iii](#_Toc6592837)

[SUMMARY iv](#_Toc6592838)

[TABLE OF CONTENTS 1](#_Toc6592839)

[LIST OFTABLES AND ILLUSTRATIONS 2](#_Toc6592840)

[CHAPTER 1 –STATISTICS FUNCTIONS 3](#_Toc6592841)

[1.1 Overview: 4](#_Toc6592842)

[1.2Functions’details: 4](#_Toc6592843)

[1.2.3 Exception: 13](#_Toc6592918)

[CHAPTER 2 – HISTOGRAM 14](#_Toc6592924)

[2.1 Overview 14](#_Toc6592925)

[2.2 Histogram Equalization report 14](#_Toc6592926)

[2.2.1 What is an Image histogram? 14](#_Toc6592927)

[2.2.2 What is Histogram equalization? 15](#_Toc6592928)

[2.2.3 Illustrative program 17](#_Toc6592940)

[2.2.4How about Color image? 20](#_Toc6592948)

LIST OFTABLES AND ILLUSTRATIONS

**LIST OF ILLUSTRATION**

Picture 1.1: mean() sample program 14

[Picture 1.2: harmonic\_mean() sample program 12](#_Toc6592857)

[Picture 1.3: median() sample program 13](#_Toc6592863)

[Picture 1.4: median\_low() sample program 14](#_Toc6592869)

[Picture 1.5: median\_high() sample program 15](#_Toc6592875)

[Picture 1.6: median\_grouped() sample program 16](#_Toc6592881)

[Picture 1.7: mode() sample program 17](#_Toc6592892)

[Picture 1.8: pstdev() sample program 17](#_Toc6592899)

[Picture 1.9.2: pvariance() sample program 18](#_Toc6592907)

[Picture 1.10: stdev() sample program 19](#_Toc6592912)

[Picture 1.11: variance() sample program 20](#_Toc6592917)

[Picture 1.12: median\_grouped() exception 20](#_Toc6592921)

[Picture 1.13: stdev() exception 21](#_Toc6592923)

[Picture 2.3 Importing libraries 24](#_Toc6603648)

[Picture 2.4 Load in and out image 24](#_Toc6603650)

[Picture 2.5 Image landscape.jpg 25](#_Toc6603652)

[Picture 2.6 Plotting step 25](#_Toc6603654)

[Picture 2.7 Original image histogram 26](#_Toc6603656)

[Picture 2.9 Histogram of equalized image and equalized image 27](#_Toc6603658)

**LIST OF TABLES**

[Table1.1Summary of Statistics Functions 2](#_Toc387689363)

**SUMMARY TABLE OF STATISTICS FUNCTIONS**

Statistics module of Python is actually a very powerful tool which allocates us some useful functions that truly help us a lot in computing anything related to Statistics. Below is a summary of all functions included in Statistics module:

|  |  |
| --- | --- |
| **AVERAGE AND MEASURES OF CENTRAL LOCATION** | |
| [mean()](https://docs.python.org/3/library/statistics.html#statistics.mean) | Return mean(average) of a list of data |
| [harmonic\_mean()](https://docs.python.org/3/library/statistics.html#statistics.harmonic_mean) | Returns the harmonic mean of an arrangement of data. |
| [median()](https://docs.python.org/3/library/statistics.html#statistics.median) | Return the middle value from an unsorted list |
| [median\_low()](https://docs.python.org/3/library/statistics.html#statistics.median_low) | Returns the low median of numeric data |
| [median\_high()](https://docs.python.org/3/library/statistics.html#statistics.median_high) | Returns the high median of numeric data |
| [median\_grouped()](https://docs.python.org/3/library/statistics.html#statistics.median_grouped) | Return the median of grouped continuous data, calculated as 50thperecentile. |
| [mode()](https://docs.python.org/3/library/statistics.html#statistics.mode) | Return the most common data point in a set of data |
| **MEASURES OF SPREAD** | |
| [pstdev()](https://docs.python.org/3/library/statistics.html#statistics.pstdev) | Return the population standard deviation |
| [pvariance()](https://docs.python.org/3/library/statistics.html#statistics.pvariance) | Returns the actual population variance of the values passed as parameter. |
| [stdev()](https://docs.python.org/3/library/statistics.html#statistics.stdev) | Returns the actual standard deviation of the values passed as parameter. |
| [variance()](https://docs.python.org/3/library/statistics.html#statistics.variance) | Returns the actual variance of the values passed as parameter. |

Table 1.1 Summary of statistics functions

CHAPTER 1 –STATISTICS FUNCTIONS

1.1 Overview:

*In this section, I am going to present the details of each function which was enumerated in Table1.1 Summary of Statistics Functions. Firstly, I would like to give out the steps I am going to conduct this report.*

* *Name of presenting specific statistics function*
* *Brief description of its function*
* *Syntax for use*
* *Input and Output*
* *Function’s formula (if necessary)*
* *Example program with source codes and result for demonstration*

1.2Functions’details:

1.2.1Average and measures of central location:

1.[mean()](https://docs.python.org/3/library/statistics.html#statistics.mean)is a Python function which can be used to calculate average of a given list of data.

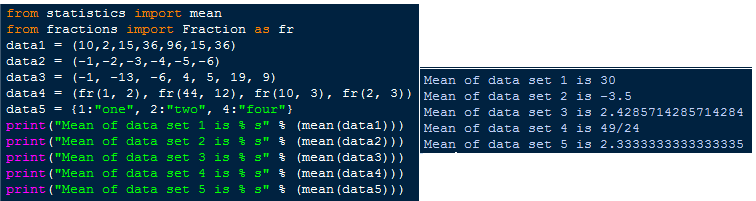
Syntax: statistics.mean(data)

Input: List or a set of numbers.

Output: Sample arithmetic average of the provided list

Formula: 

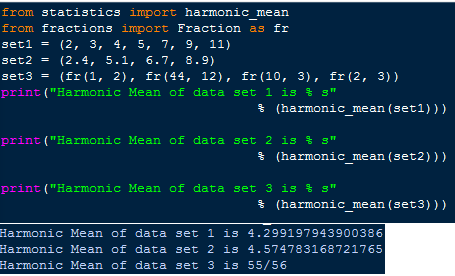
Example:



Picture 1.1: mean() sample program

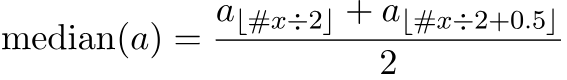
2.[harmonic\_mean()](https://docs.python.org/3/library/statistics.html#statistics.harmonic_mean) help us return the contrary mean of a set of data which is a list or tuple or iterator of real valued numbers. Moreover, harmonic mean is calculated only using positive values in list, set or any sequence.

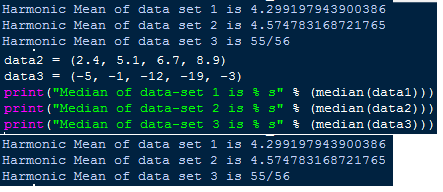
* Syntax: statistics.harmonic\_mean(data)
* Input: List or a set of positive numbers.
* Output: *Returns the harmonic\_mean of the given set of data.*
* Formula
* Example:



Picture 1.2: harmonic\_mean() sample program

3. [median()](https://docs.python.org/3/library/statistics.html#statistics.median)function helps us calculate median value from an unsorted data list. Moreover, the input is a list of data which is not necessary to be sorted and for a dataset, this could be came up with the middle value. For more specific, if the given set of data is an odd set, median value can be highly recommended as the middle one,else if that set is an even one, median value should be the average scale of two middle values.

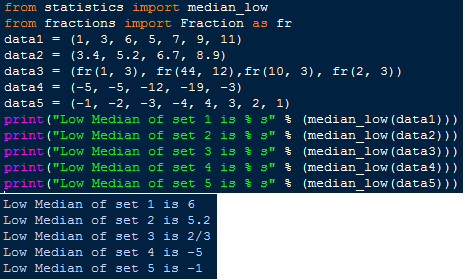
* Syntax: statistics.median(data)
* Input: List or tuple or an iterable with a set of sorted numeric values, if the given set is not sorted yet, this function aims to sort it.
* Output: Returns the harmonic\_mean value of the set. With odd set returns middle value, with even set returns mean of two middle values.
* Formula: 
* Example:



Picture 1.3: median() sample program

4.[median\_low()](https://docs.python.org/3/library/statistics.html#statistics.median_low)function helps us return the low median value of a set of data. This time, we will be confused by the results between [median\_low()](https://docs.python.org/3/library/statistics.html#statistics.median_low)and [median()](https://docs.python.org/3/library/statistics.html#statistics.median)but let not be worried from this point, median low value is always the member of the set of data and this value is definitely different from the median value which was presented above, if the set is an odd set returns the middle value, if it is an even set the smaller of the two middle values is returned.

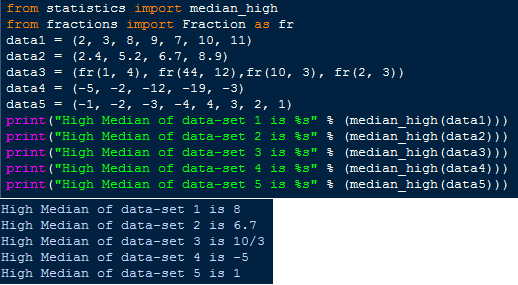
* Syntax: statistics.median\_low (data)
* Input: List or tuple or an iterable with a set of sorted numeric values, if the given set is not sorted yet, this function aims to sort it.
* Output: Returns the median\_low value of the set. With odd set returns middle value, with even set returns the smaller value of two middle values.
* Formula: *none*
* Example:



Picture 1.4: median\_low() sample program

5. [median\_high()](https://docs.python.org/3/library/statistics.html#statistics.median_high) function will return the high median value of a set of data. Alike [median\_low()](https://docs.python.org/3/library/statistics.html#statistics.median_low), this function tends to return the middle value of the given set if this is an odd set, but if this is an even set, this function will return the larger value of two middle values of the set.

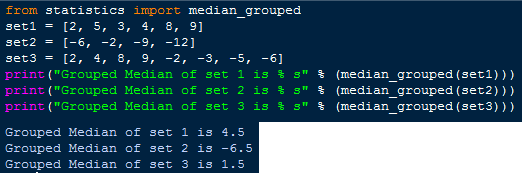
* Syntax: statistics.median\_high (data)
* Input: List or tuple or an iterable with a set of sorted numeric values, if the given set is not sorted yet, this function aims to sort it.
* Output: Returns the median\_high value of the set. With odd set returns middle value, with even set returns the larger value of two middle values.
* Formula: *none*
* Example:



Picture 1.5: median\_high() sample program

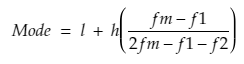
6.[median\_grouped()](https://docs.python.org/3/library/statistics.html#statistics.median_grouped)function allows you to calculate median of grouped and continuous set of data. Besides, list or tuple or an iterable with a set of numeric values interval(default by 1)**:** Determines the width of grouped data and changing. It will also change the interpolation of calculated median.

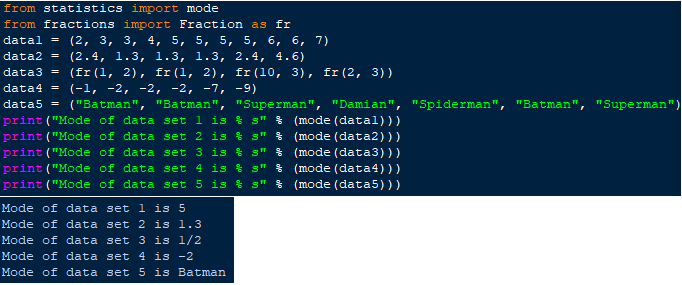
* Syntax: statistics.median\_grouped (data, interval = 1)
* Input: List or tuple or an iterable with a set of sorted numeric valuesinterval (default by 1): Determines the width of grouped data and changing. It will also change the interpolation of calculated median.
* Output: Return the median of grouped continuous data, calculated as 50th perecentile.
* Formula: median = L + interval \* (N / 2 - CF) /F
* Example:



Picture 1.6: median\_grouped() sample program

7. [mode()](https://docs.python.org/3/library/statistics.html#statistics.mode)function will return the most presented value in the given set of data

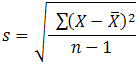
* Syntax: statistics.mode(data)
* Input:  Tuple, list or an iterator of real value or it can be a string.
* Output: Return the most appeared data point from the given set of data or a string.
* Formula:  *, where* 
  + - * *L: Lower boundary of modal class*
      * *H: size of model class*
      * *Fm: Frequency corresponding to modal class*
      * *F1: Frequency preceding to modal class*
      * *F2: Frequency proceeding to modal class*
* Example:

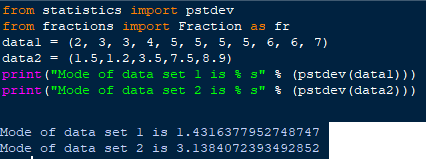


Picture 1.7: mode() sample program

1.2.2 Measures and Spread:

8. pstdev()functions will return the population standard deviation **(**square root of population variance**)** of the data.

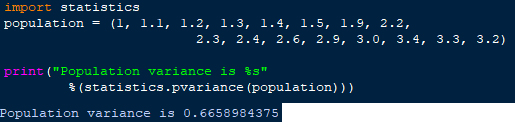
* Syntax: statistics.pstdev(data, mu=None)
* Input:  A list of data.
* Output: Return the standard population standard deviation
* Formula: 
* Example:



Picture 1.8: pstdev() sample program

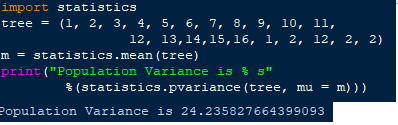
9. [pvariance()](https://docs.python.org/3/library/statistics.html) function helps calculate the variance of an entire data set rather than that of a sample.

* Syntax: statistics.**pvariance**(data, mu=None)
* Input:  A set of data or an iterable with real valued numbers, when mu is needed of some defined cases, it should be the actual mean of data-set/ population as value.
* Output:Returns the population variance of the values passed as parameter.
* Formula:
* Example:
  + - * Case without “mu”:



Picture 1.9.1: pvariance() sample program

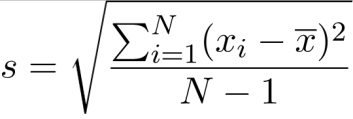
* + - * Case with “mu”:

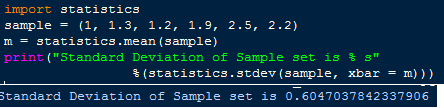


Picture 1.9.2: pvariance() sample program

10. [stdev()](https://docs.python.org/3/library/statistics.html)function allows us to calculate the standard deviation of a set of data

- Syntax: statistics.**stdev**(data list,xbar)

* Input:  An iterable with real valued numbers and xbar(optional) is to take actual mean of data set as a value.
* Output:Returns the actual standard deviation of the values passed as parameter.
* Formula: 
* Example:

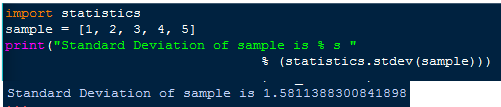


Picture 1.10: stdev() sample program

11.[variance()](https://docs.python.org/3/library/statistics.html)function helps to calculate the variance from a sample of data (sample is a subset of populated data). Besides, we should use this function when variance of a sample needs to be calculated. Moreover, there is a funciton named pvariance() which returns variance of entire set of data, because of this point, pvariance and variance are different.

- Syntax: statistics.**variance**(data,xbar)

* Input:  An iterable with real valued numbers and xbar(optional) is to take actual mean of data set as a value.
* Output:Returns the actual standard deviation of the values passed as parameter.
* Formula: 
* Example:

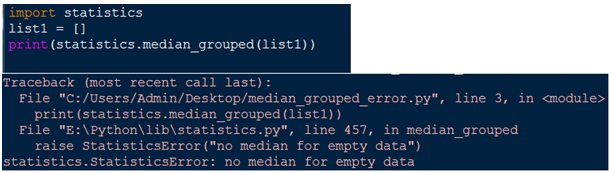


Picture 1.11: variance() sample program

1.2.3 Exception:

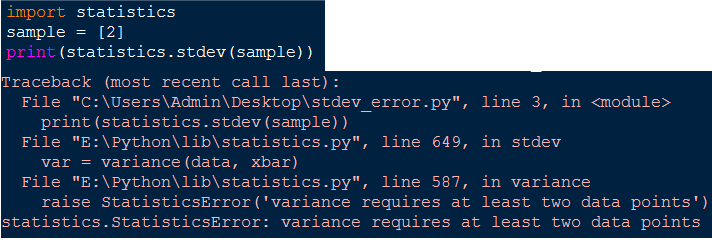
Last but not least, some of those statistics functions come with statistical exception, when this is raised, Statistics\_Error will be thrown. For more details, this type of error is being thrown when an operation of the functioning function receives an argument that has the right type but an inappropriate value. I am going to illustrate some of statistics functions that would throw statistics error when exception is raised.

Example1, in function median\_grouped(), StatisticsError is raised when passed iterable is empty or when list is null.



Picture 1.12: median\_grouped() exception

Example 2, in stdev() function , statistics error will raise when the date set does not contain enough data points



Picture 1.13: stdev() exception

CHAPTER 2 – HISTOGRAM

2.1 Overview

In this chapter, I am going survey graphical equalizing method called Histogram Equalization. Specifically, this chapter will be designed to go through these steps:

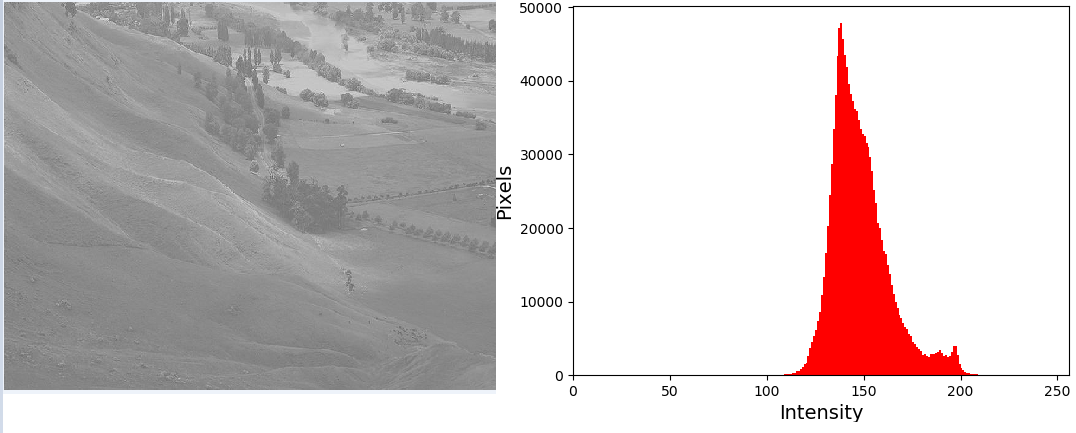
* What is an Image Histogram?
* What is Histogram equalization?
* Illustrative program?
* Overview for color histogram

2.2 Histogram Equalization report

2.2.1 What is an Image histogram?

* It is a type of histogram which acts as graphical representation of the intensity distribution of an digital image.
* It quantifies the number of [pixels](https://en.wikipedia.org/wiki/Pixels) for each tonal value considered.
* In real life, image histograms are presented on many modern [digital cameras](https://en.wikipedia.org/wiki/Digital_cameras). Photographers can consume these histograms as a tool to show the distribution of graphical captured, and whether image detail has been lost to elaborate highlights or extinguished shadows.However, this is known to be less useful when using a [raw image format](https://en.wikipedia.org/wiki/Raw_image_format), as the [dynamic range](https://en.wikipedia.org/wiki/Dynamic_range#Photography) of the displayed image may only be an approximation to that in the raw file.

For instance, below is an image histogram:



Picture 2.1: Image Histogram

2.2.2 What is Histogram equalization?

Briefly, histogram equalization is a method in a method in image processing that enhance the contrast in an image by using image histogram, which is about to stretch out the intensity range. Let take look at this

Not to waste some time, I am going to present the details of histogram equalization algorithm according to the book “Digital Image Processing” by R. C. Gonzalez and R. E. Woods.

Let *f* be a given image occured as a *mr* by *mc* matrix of integer pixel intensities ranging from 0 to *L − 1*. *L* is the number of possible intensity values, often 256. Let *p* indicate the normalized histogram of *f* with a bin for each possible intensity. Which mean:

 n = 0, 1, ..., L − 1.

The histogram equalized image g will be defined by

, (1)

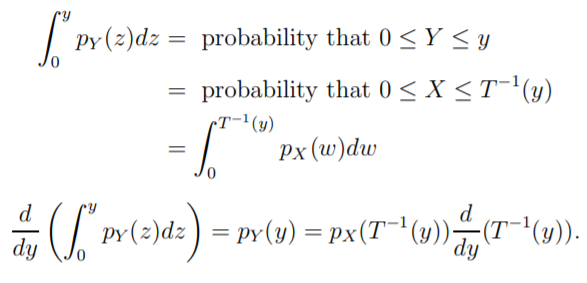
where floor() rounds down to the nearest integer. This is equivalent to transforming the pixel intensities, k, of f by the function

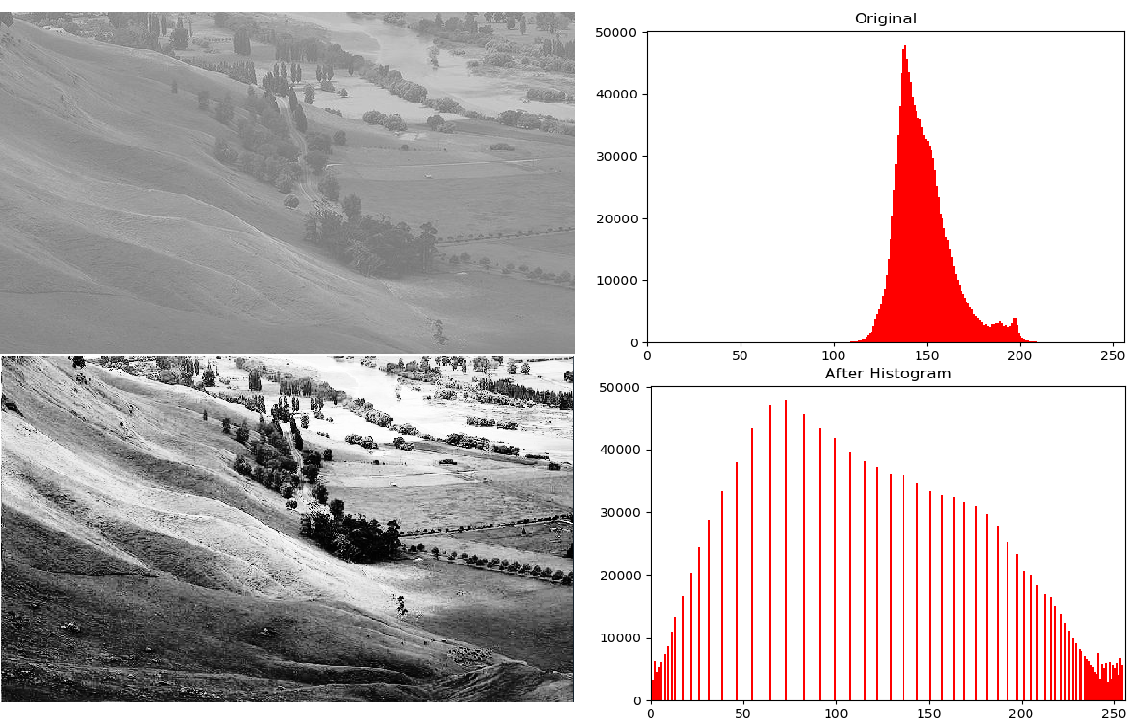
.

The motivation for this transformation comes from thinking of the intensities of f and g as continuous random variables X, Y on [0, L − 1] with Y defined by

 (2)

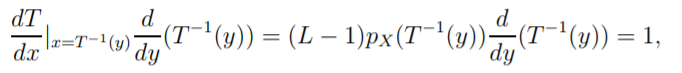
where *px* is the probability density function of f. T is the cumulative distributive function of X multiplied by (L − 1). Assume for simplicity that T is differentiable and invertible. It can then be shown that Y defined by T(X) is uniformly distributed on [0, L − 1], namely that pY (y) = 1/(L−1) .





Picture 2.2 Histogram equalization applied to low contrast image

Note that  so

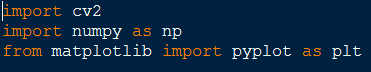


Which means,  .

Our discrete histogram is an approximation of pX (x) and the transformation in Equation (1) approximates the one in Equation (2). While the discrete version won’t result in exactly flat histograms, it will flatten them and in doing so enhance the contrast in the image. The result of applying Equation (1) to the landscape.jpg test image is shown in Picture 2.2.

2.2.3 Illustrative program

Firstly, we need to import necessary library for histogram equalization. In my source codes, I used opencv-python, numpy for image processing and histogram equalizing, matplotlib for plotting charts and histogram scaling.



Picture 2.3 Importing libraries

Secondly, I am going load in image and load it out at the same time to show the original piece of work



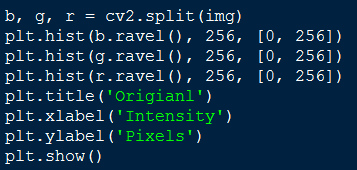
Picture 2.4 Load in and out image

which we will receive this



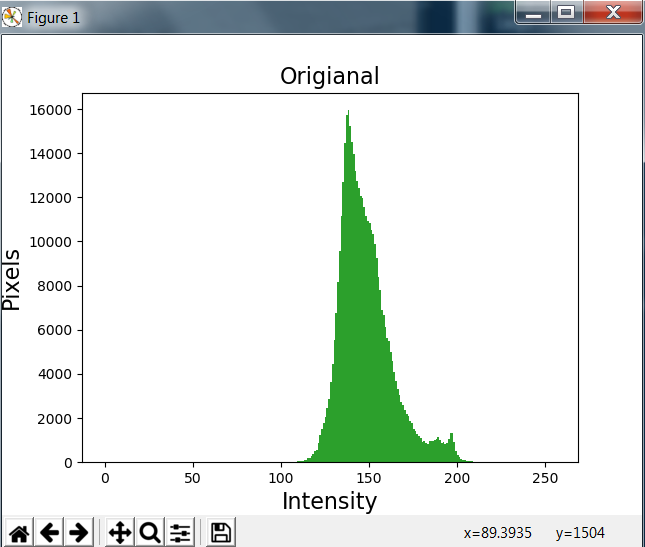
Picture 2.5 Image landscape.jpg

Thirdly, I plot a histogram for this image



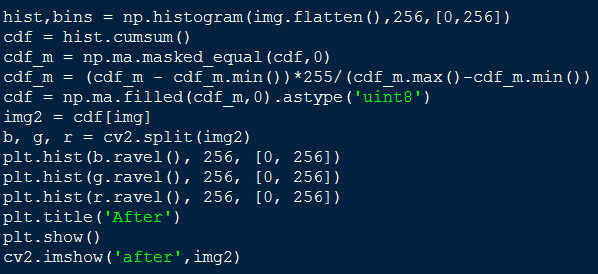
Picture 2.6 Plotting step

Which we will receive below image histogram

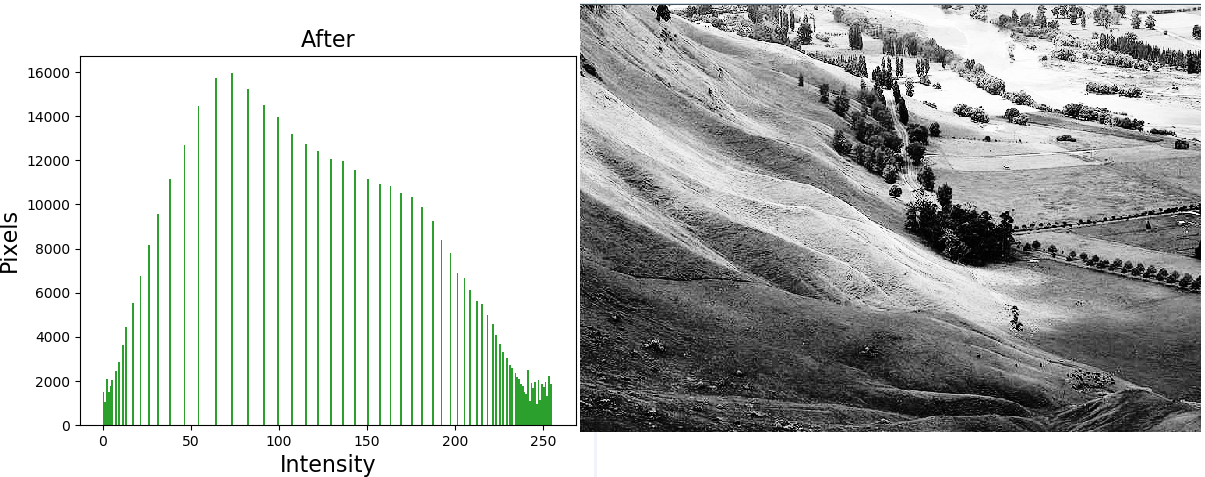


Picture 2.7 Original image histogram

Next up, I use histogram equalization method to equalize my image then I plot the histogram after equalization and the processed image which is img2. Finally, we can notice some differences between the previous and after histogram equalization, clearly, the intensity range now is stretched and perceived with a sporadic distribution which comes with the after tonal changed image is more defined and simply beautiful.



Picture 2.8 Histogram equalizing step



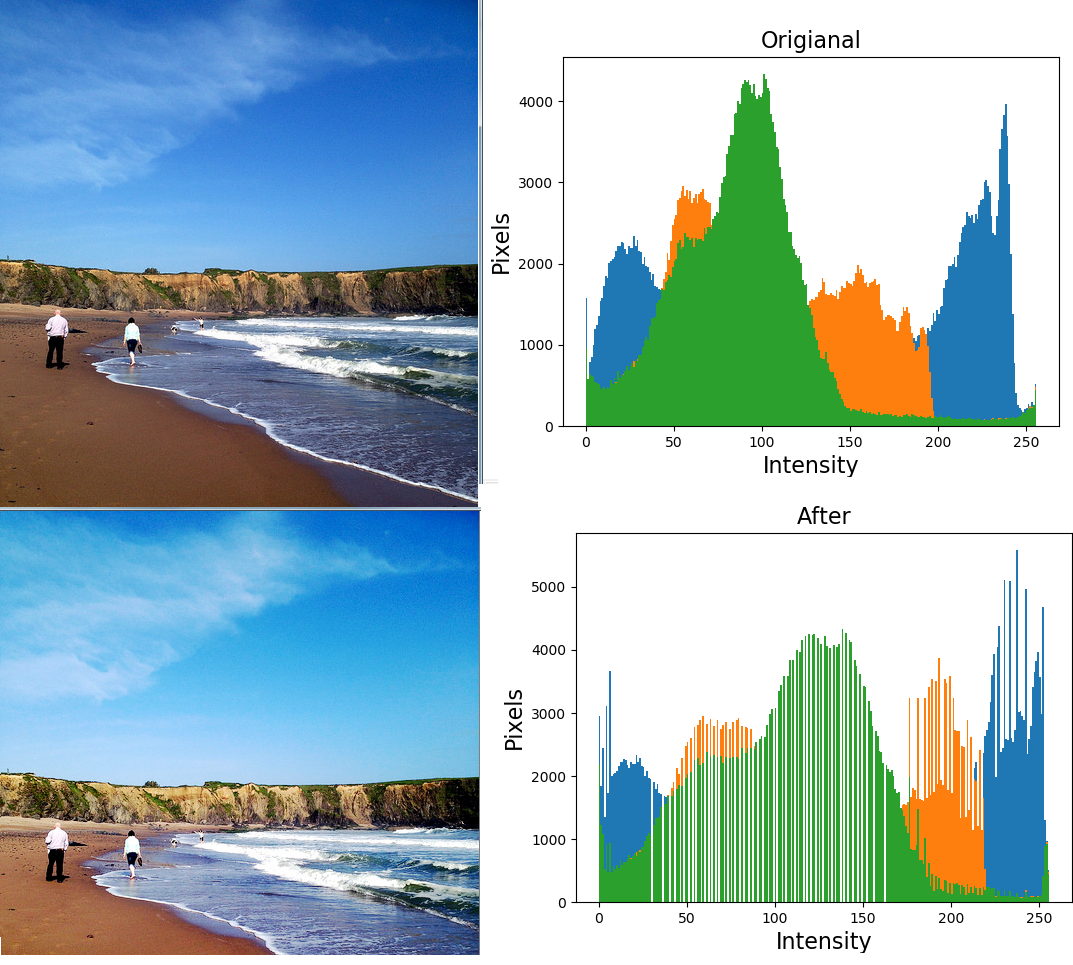
Picture 2.9 Histogram of equalized image and equalized image

2.2.2 How about color images?

The answer is “YES”, we can apply histogram equalization to color images by using three-dimensional spaces like [RGB](https://en.wikipedia.org/wiki/RGB_color_space) or [HSV](https://en.wikipedia.org/wiki/HSV_color_space).

The formation of a color histogram is rather simple. We can simply count the number of pixels for each 256 scales in each of the 3 RGB channel, and plot them on 3 individual bar graphs. Specifically, a color histogram is based on a certain color space, such as RGB or HSV. When we compute the pixels of different colors in an image, if the color space is large, then we can first divide the color space into certain numbers of small intervals. Each of the intervals is called a bin. This process is called color quantization. Then, by counting the number of pixels in each of the bins, we get the color histogram of the image.

Let use my source codes to do it as an example, this time I would like to use a digital color image named beach.jpg and we will receive the result as below



Picture 2.10 Color image equalizing result

As we can observe, each color channel is now equalized.

**REFERENCES**

1. <https://docs.python.org/3/library/statistics.html>
2. [https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)
3. <https://docs.opencv.org/2.4.13.7/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html>
4. <https://en.wikipedia.org/wiki/Histogram_equalization>.
5. R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third Edition, 2008.
6. <https://en.wikipedia.org/wiki/Image_histogram>.
7. <https://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf>.
8. <https://en.wikipedia.org/wiki/Color_histogram>.