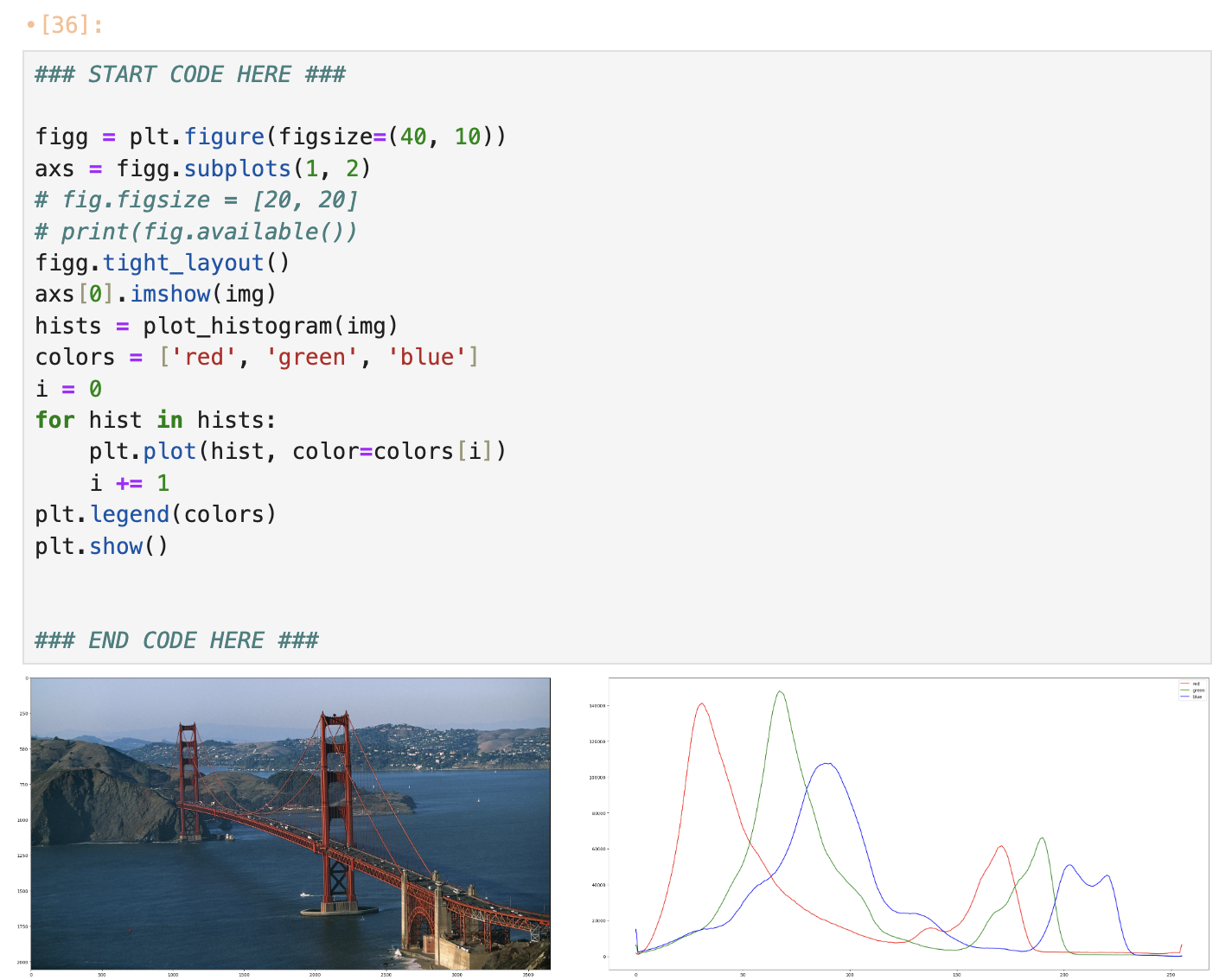
**Lab2-2 : Image Enhancement with Statistical Operation**

**Histogram Equalization**

****

* Return ออกมาเป็น List ของ Histogram ที่มีขนาดเป็น 3 โดยจะเป็น Histogram ของสีแดง สีเขียว สีฟ้า (RGB) ตามลำดับ
* ใช้ฟังชั่น cv2.calcHist() โดยจะรับ parameter เป็น image และ index ของสีที่สนใจ
* นอกจากนี้ยังรับ mask ที่เราไม่ได้ต้องการใช้ และก็เป็น size และ range ของ dimension สีของรูปภาพ

**Before:**

****

* นำ list ของ Histogram data จากฟังชั่นข้างต้นมาแสดงทีละอัน โดยใช้สีที่แตกต่างกัน โดยใช้ฟังชั่น plt.ploy(hist, color=colors[i]) โดย hist คือ iterator ของ list และ colors = [‘red’, ‘green’, ‘blue’ ]

**After:**

**A screenshot of a computer

Description automatically generated**

* นำรูปภาพเดิม(ซึ่งเป็น array 3 มิติ) มาแยกเป็น array 2 มิติจำนวน 3 อัน เป็นของ channel R, G และ B ตามลำดับ จากนั้นนำแต่ละ Channel มาเข้าเป็นพารามิเตอร์ของฟังก์ชัน cv2.equalizeHist() เพื่อกระจายเฉดสีแต่ละสี

**Image Histogram Matching**

**Load image:**

**A white background with red and blue text

Description automatically generated**

* โหลดรูปภาพโดยใช้ฟังชั่น cv2.imread() แล้วนำมา convert จากที่ได้เป็น BGR ให้เป็น RGB ด้วยฟังก์ชัน cv2.cvtColor()

**Display Histogram matching:**

**A screenshot of a computer

Description automatically generated**

* ในการ display PDF เราสามารถ plot ได้โดยตรงจาก histrogram data ข้อก่อนหน้าได้เลย แต่ว่าในแนวแกน y เราต้องเป็นความน่าจะเป็น เราจึงต้องหารผลรวมของ histogram นั้นด้วย
* ส่วนในการ display CDF เราจำเป็นต้องนำมาเข้าฟังชั่น cumsum() เพื่อซึ่งฟังก์ชันนี้จะ return ออกมาเป็น list ใหม่ซึ่งคำนวณออกมาเป็น cumulative แล้ว ยกตัวอย่างเช่นหาก input เป็น [1,2,3,4] จะได้ output เป็น [1, 3, 6, 10]

**Compare with Equalized Image**

**A screenshot of a computer

Description automatically generated**

* จากข้อนี้ก็ใช้หลังการเดียวกันกับข้อก่อนหน้า แต่เราต้องแก้การพล็อตของเราในฟังก์ชัน subplot ให้เป็น 3 แถวด้วย แล้วก็ทำแบบเดียวกันกับข้อก่อนหน้าได้เลย