**LẬP CHỈ MỤC BOOLEAN VỚI NUMPY**

Contents

[1. Đọc file CSV với numpy 1](#_Toc115943090)

[2. Mảng Boolean 5](#_Toc115943091)

[3. Đánh dấu Boolean với mảng 1 chiều 8](#_Toc115943092)

[4. Đánh dấu Boolean với mảng 2 chiều 12](#_Toc115943093)

[5. Gán giá trị trong mảng 14](#_Toc115943094)

[6. Gán giá trị sử dụng mảng Boolean 18](#_Toc115943095)

[7. Bài toán đâu là sân bay được đến nhiều nhất ? 21](#_Toc115943096)

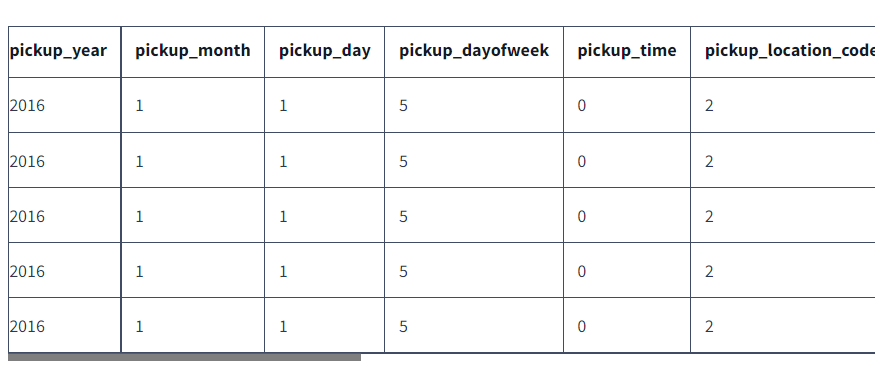
[8. Bài toán tính dữ liệu các chuyến đi trong một dữ liệu sạch 23](#_Toc115943097)

|  |  |
| --- | --- |
| Tên | MSSV |
| Trần Nguyễn Quang Lâm | 20139040 |
| Tô Gia Huy | 20139003 |

# Đọc file CSV với numpy

Kỹ thuật mới: Boolean Indexing(lập chỉ mục boolean).

VD :5 Hàng đầu tiên của data:



Bắt đầu học cách sử dụng hàm numpy.genfromtxt() để đọc files vào NumPy ndarrays. Đây là cú pháp đơn giản cho hàm, và giải thích cho 2 thông số:

np.genfromtxt(filename, delimiter = None)

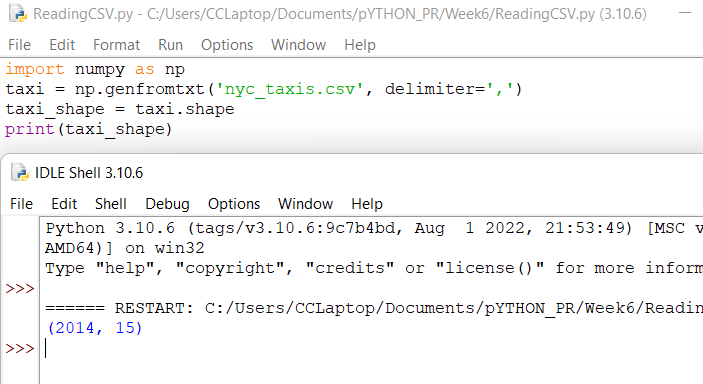
* filename: thường là string sẽ đại diện cho đường dẫn đến file để đọc.
* delimiter: ký tự đặc biệt, xác định phần tử chia string ra mỗi giá trị.

Cách để đọc file tên data.csv:

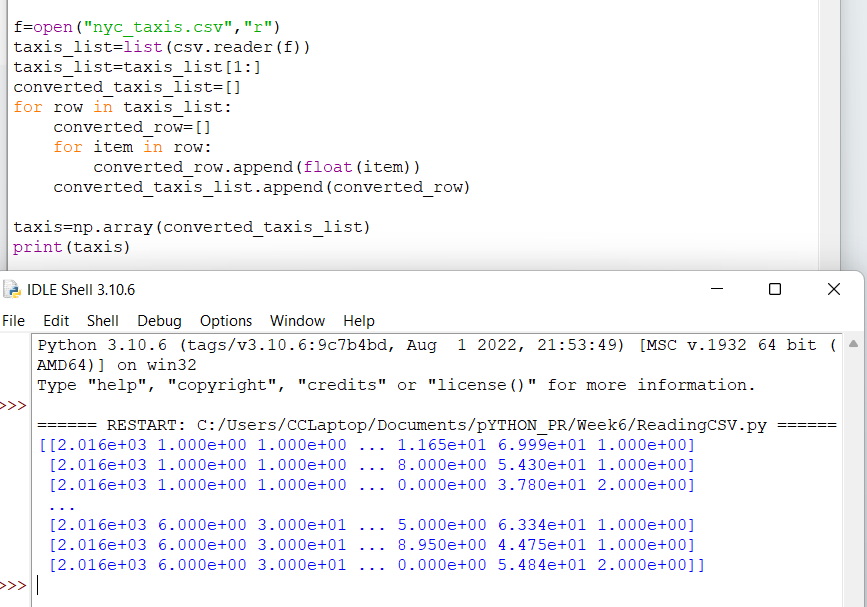
data = np.genfromtxt(‘data.csv’, delimiter =’,’)

hãy đọc file nyc\_taxis.csv vào NumPy:

* Đầu tiên import thư viện numpy lưu dưới tên np
* Dùng lệnh numpy.genfromtxt() để đọc file nyc\_taxis.csv vào numpy, gán cho biến taxi
* Dùng attribute ndarray.shape để gán kích thước của taxi vào taxi\_shape
* In ra màn hình kết quả kích thước của mảng

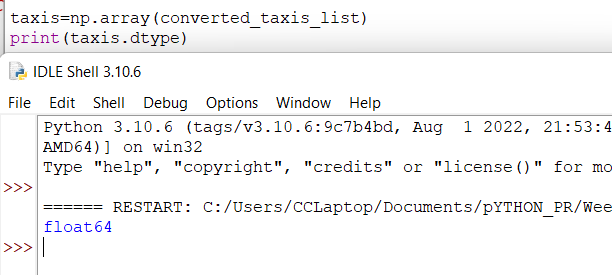


Chuyển dữ liệu trong file csv thành dạng list của các list



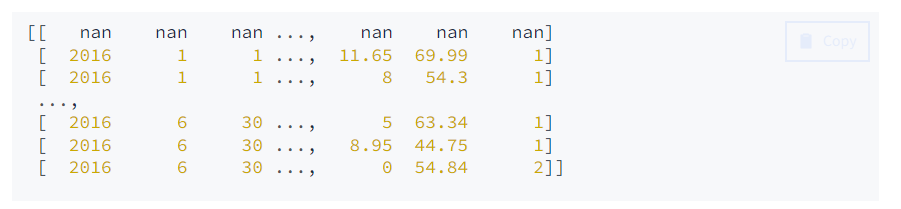
*Lưu ý rằng trước khi chuyển list của list thành mảng n chiều, ta phải chuyển các phần tử về dạng float. Lí do là vì mảng n chiều của Numpy chỉ có thể chứa một kiểu dữ liệu*

Dùng attribute ndarray.dtype để biết kiểu dữ liệu của mảng taxis là float 64



Numpy dùng kiểu float 64 vì nó cho phép đọc hầu hết các giá trị từ file CSV, kiểu float64 này sẽ tương đồng với float trong python, có số 64 vì 64 là số bit nó dùng để lưu 1 giá trị

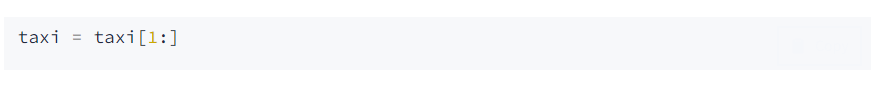
Nếu ta coi kết quả từ bài tập trước,ta thấy taxi chứa hầu hết là số nhưng vẫn có các giá trị Nan



Nan viết tắt cho Not A Number-có nghĩa là giá trị không được lưu dưới dạng số, và thường có giá trị Null, giống với None của Python

Ta thường thấy NaN khi một giá trị bị mất, nhưng trong trường hợp này ta có giá trị NaN vì dòng đầu tiwn trong file CSV chứa tên các cột, numpy không chuyển đổi các giá trị string này như pickup\_year về dạng float64 được

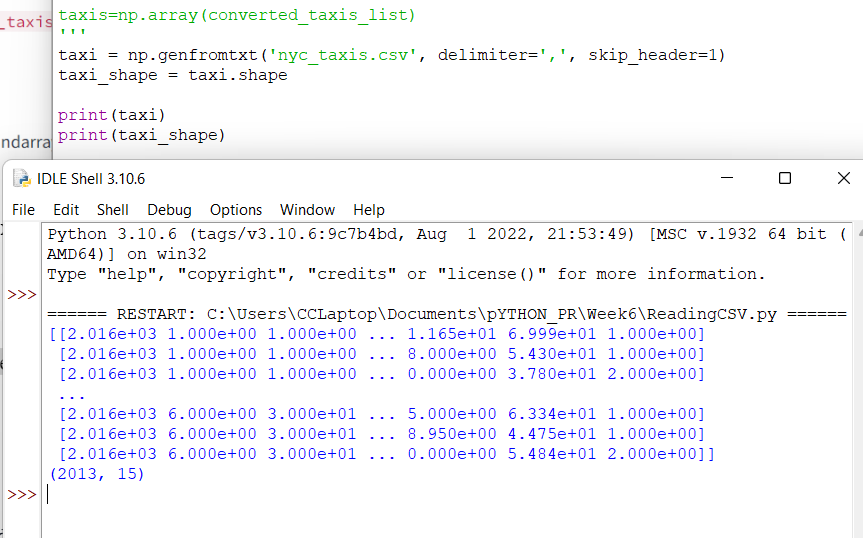
Bỏ các hàng header khỏi mảng:



Ta có thể dùng cách khác, truyền vào đố số skip\_header cho hàm numpy.genfromtxt(). Đối số skip\_header chấp nhận một số nguyên, bắt đầu chạy từ hàng đầu tiên cho tới hàng của số đã nhập thì bỏ.

*Lưu ý rằng đây là số hàng không phải đánh dấu, nên hàng đầu tiên bắt đầu bằng số 1*

* Dùng hàm numpy.genfromtxt() để đọc file nyc\_taxis.csv, bỏ hàng đầu , gán kết quả vào taxi
* Gán kích thước của taxi vào taxi\_shape
* xuất ra kết quả biến taxi và kích thước của nó



# Mảng Boolean

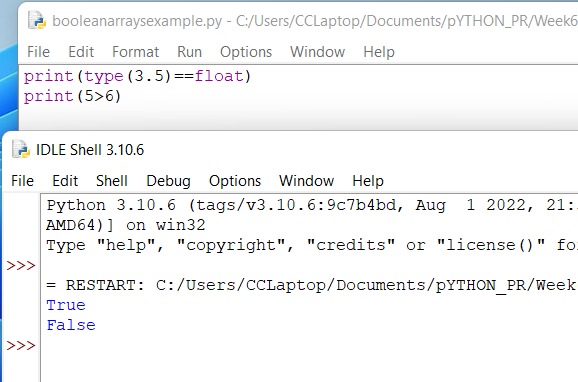
Phương pháp hiệu quả và hay được sử dụng nhất để đánh dấu data là sử dụng **mảng Boolean**. Mảng Boolean là một mảng mà các phần tử của nó có giá trị Boolean, đôi khi ta có thể gọi là v**ector Boolean** hoặc **mặt nạ Boolean**

Các giá trị Boolean(kiểu bool trong Python) có 2 giá trị sau

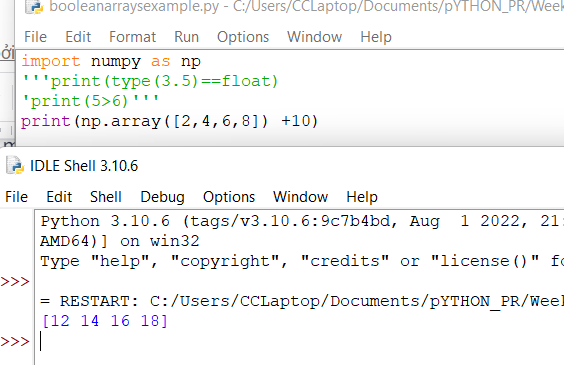
* True
* False

Dùng giá trị Boolean trong Python với các toán tử so sánh ==, >, <,!=

Ví dụ:

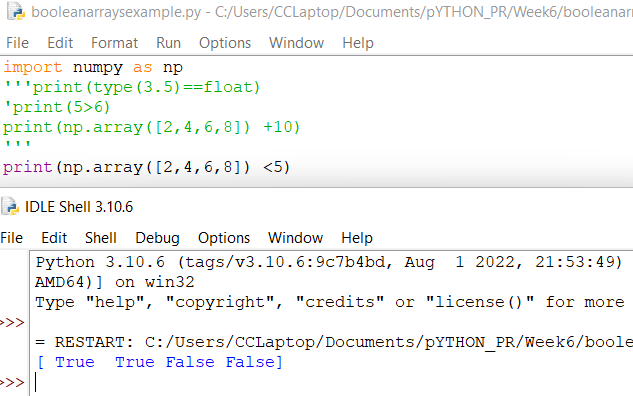


Khi cộng một mảng n chiều với một phần tử ta được một mảng n chiều mới



Toán tử +10 được áp dụng cho tất cả các phần tử

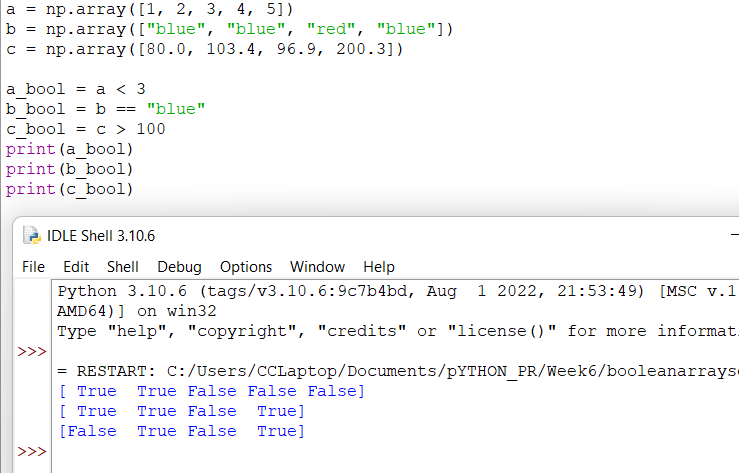
Khi ta thực hiện các toán tử Boolean giữa mảng n chiều và một phần tử duy nhất:



Tương tự như phép cộng, từng phần tử được so sánh với 5 và xuất ra kết quả True hoặc False

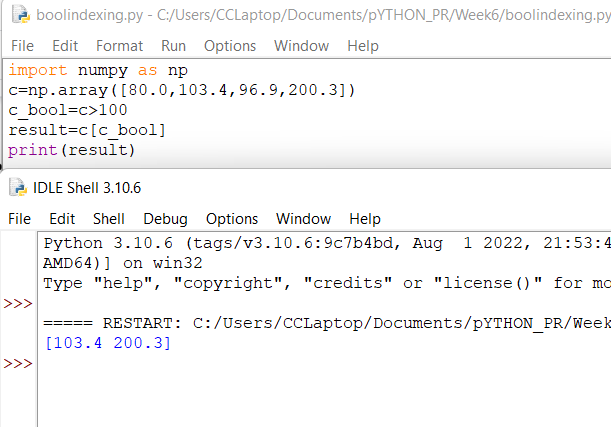
Dùng các toán tử Boolean được vector hóa để tạo nên các mảng Boolean

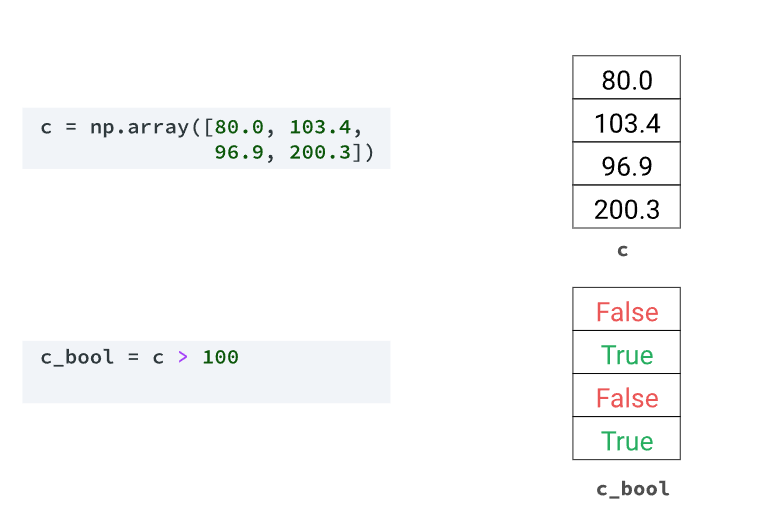
* Kiểm tra các phần tử a nhỏ hơn 3 không, gán kết quả vào a\_bool
* Kiểm tra các phần tử b có bằng giá trị “blue” không, gán kết quả vào b\_bool
* Kiểm tra các phần tử c có lớn hơn 100 không, gán kết quả vào c\_bool



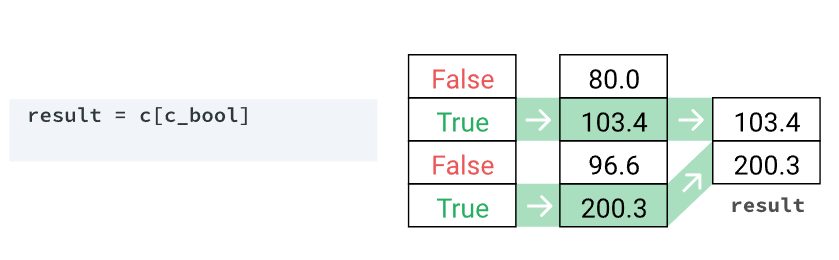
# Đánh dấu Boolean với mảng 1 chiều

Dưới đây là ví dụ đánh dấu Boolean với mảng 1 chiều





Để đưa ra các số đã lọc, ta đưa biến vào dấu ngoặc vuông như index để xuất ra thứ ta cần.



Mảng Boolean hoạt động như một bộ lọc để các giá trị tương ứng với True trở thành một phần của kết quả và các giá trị tương ứng với False sẽ bị loại bỏ.

Hãy sử dụng Boolean index để xác nhận số lượt đi taxi trong tập dữ liệu của chúng tôi từ tháng 1. Đầu tiên, hãy chỉ chọn cột pickup\_month, là cột thứ hai trong ndarray:



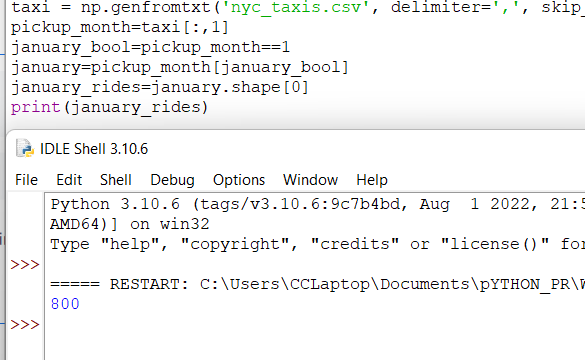
Sau đó dùng toán tử Boolean để tạo một mảng Boolean với giá trị 1 phản ánh tháng 1(tháng Giêng)



Dùng mảng boolean đã tạo này để chọn ra các giá trị trong pickup\_month tương ứng với giá trị 1:

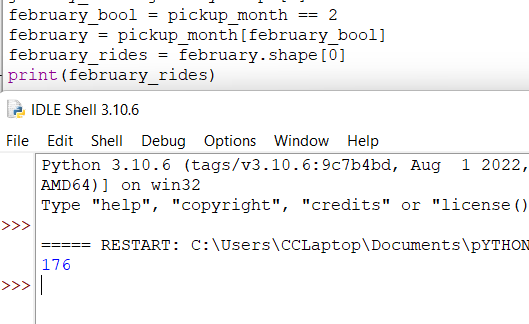


Cuối cùng dùng attribute .shape để xác định có bao nhiêu phần tử trong mảng january, tương đương vơi số lần taxi lái trong tháng 1. Dùng [0] để tách giá trị trả về từ tuple của .shape



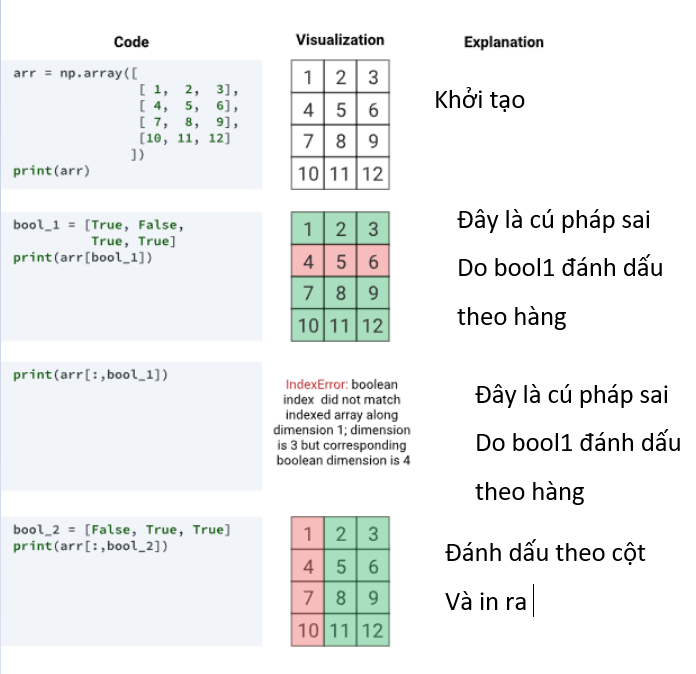
Tương tự, ta có thể tính được số cuốc taxi đã chạy trong tháng 2(February)

* Tạo mảng Boolean february\_bool tính các phần tử trong pickup\_month bằng 2
* Dùng february\_bool để lựa ra các giá trị bằng 2, gán vào kết quả february
* Dùng ndarray.shape để tính toán số cuốc chạy trong tháng 2, gán kết quả vào february\_rides, xuất ra màn hình

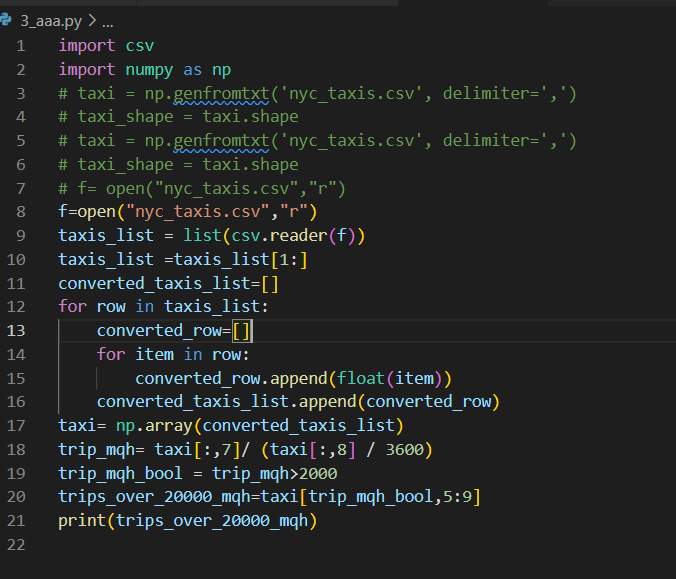


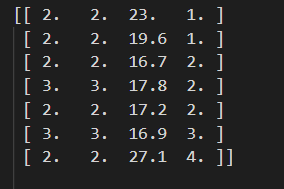
# Đánh dấu Boolean với mảng 2 chiều

Khi làm việc với ndarrays 2D, bạn có thể sử dụng lập chỉ mục Boolean kết hợp với các phương pháp lập chỉ mục khác. Hạn chế duy nhất là mảng Boolean phải có cùng độ dài với thứ nguyên bạn đang lập chỉ mục. Hãy xem một số ví dụ:



Vì mảng Boolean không chứa thông tin về việc tạo ra nó, chúng ta có thể sử dụng một mảng Boolean được tạo từ chỉ một cột của mảng của chúng ta để lập chỉ mục toàn bộ mảng

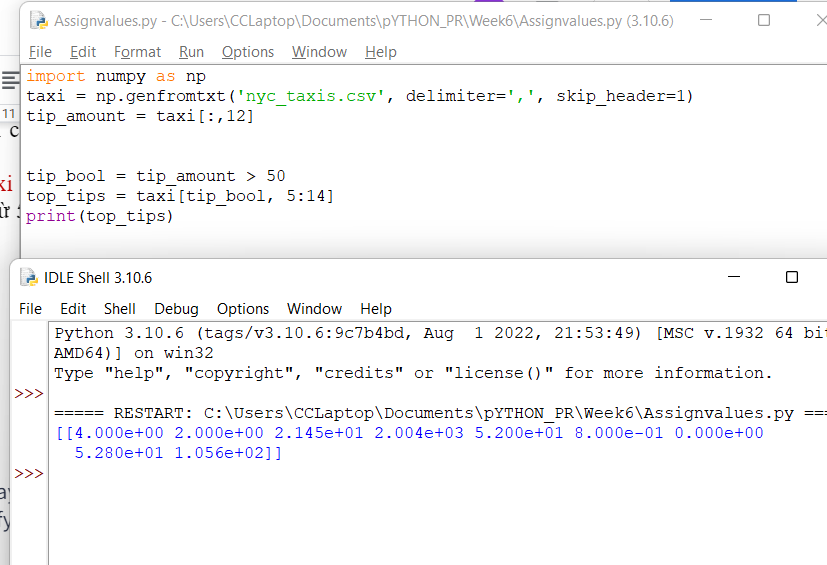




Chúng ta có thể thấy từ cột cuối cùng rằng hầu hết trong số này là những chuyến đi rất ngắn - tất cả đều có giá trị trip\_length từ 4 giây trở xuống, không dung hòa với khoảng cách chuyến đi, tất cả đều dài hơn 16 dặm

Hãy sử dụng kỹ thuật này để kiểm tra các hàng có giá trị cao nhất cho cột tip\_amount

1. Tạo mảng Boolean, tip\_bool, việc đó xác định cột nào sẽ chứa giá trị cho biến tip\_amount của hơn 50 cột.
2. Sử dụng tip\_bool để chọn tất cả các hàng từ taxi với các giá trị tip amounts của hơn 50 cột, và các cột đó sẽ được đánh số từ 5 tới 13. Gán cho mảng kết quả top\_tips

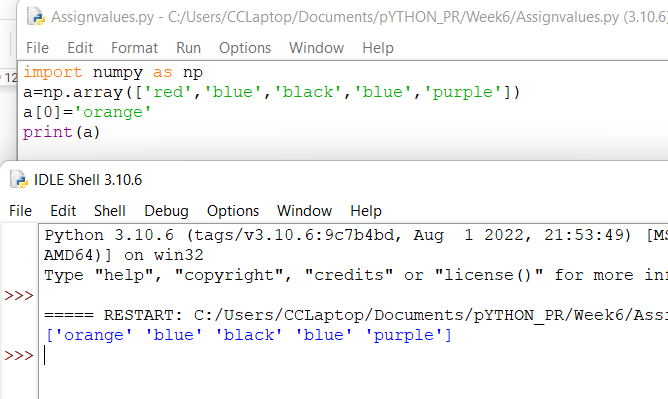


# Gán giá trị trong mảng

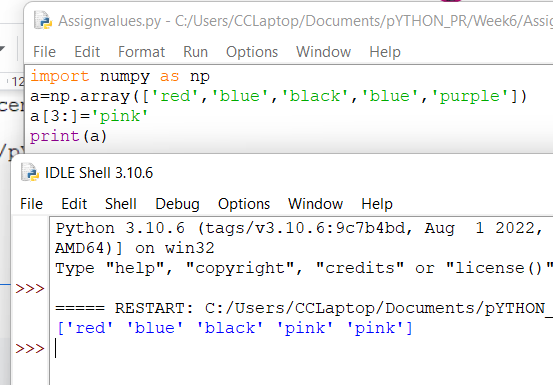
Dùng kỹ thuật lập chỉ mục (boolean indexing) đã học để điều chỉnh giá trị trong mảng, cú pháp câu lệnh (mã giả) như sau:

*ndarray[vị\_trí\_của\_giá\_trị]=giá\_trị\_mới*

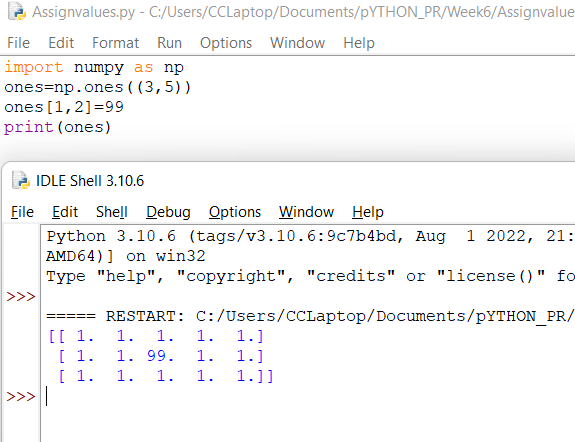
Áp dụng vào code ta có:



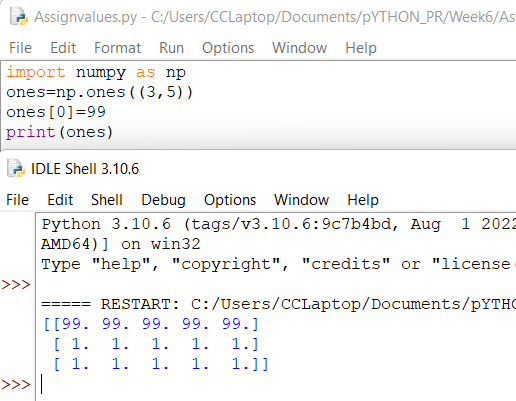
Hoặc gán một lần nhiều giá trị:



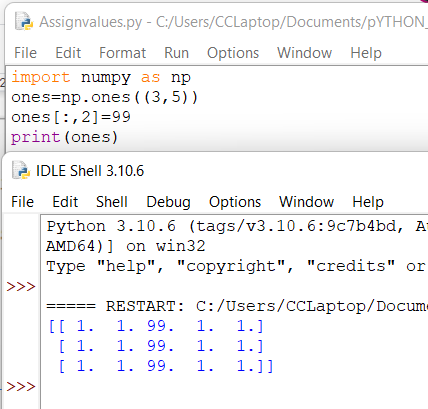
Với mảng 2 chiều cũng tương tự như mảng 1 chiều



Ta cũng có thể gán cả một hàng

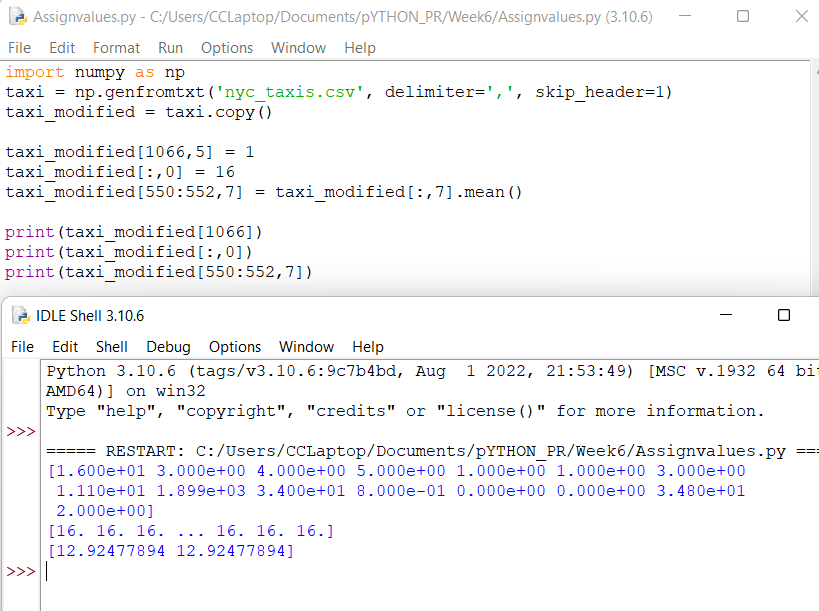


Hoặc gán cả một cột



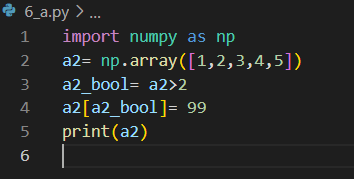
Để luyện tập với các mảng dữ liệu, ta có thể tạo ra mảng copy với hàm ndarray.copy()

* Gía trị tại cột 5 của hàng 1066 không chính xác, đổi giá trị này về 1 trong mảng copy đã tạo
* Cột đầu tiên(index 0) chứa giá trị 4 chữ số trong định dạng năm, đổi về định dạng 2 chữ số trong mảng copy đã tạo ra
* Gía trị trong cột index 7(trip\_distance) của hàng index 550 và 551 không chính xác, đổi nó về giá trị mean của cột đó trong mảng copy đã tạo ra



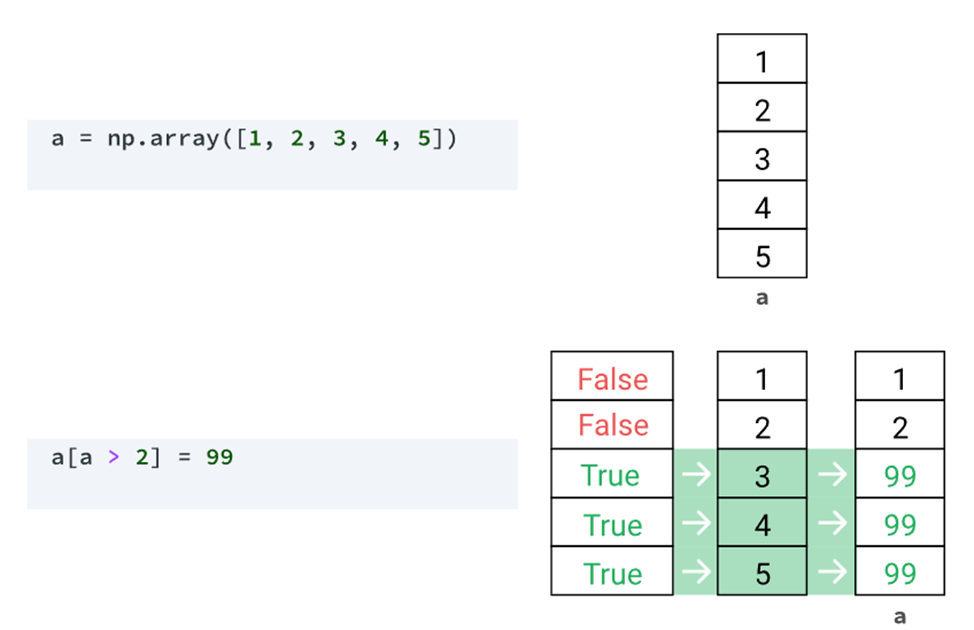
# Gán giá trị sử dụng mảng Boolean

Mảng Boolean trở nên rất mạnh mẽ khi chúng ta sử dụng chúng để gán. Hãy xem một ví dụ:





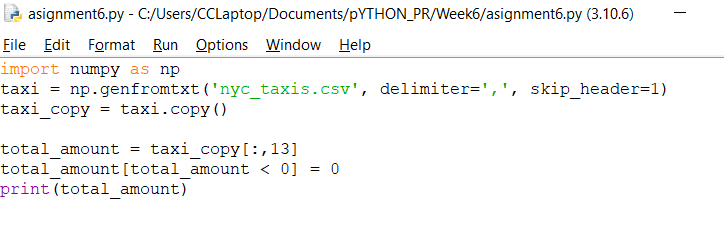
Mảng Boolean điều khiển các giá trị mà gán áp dụng và các giá trị khác không thay đổi. Hãy xem cách mã này hoạt động:

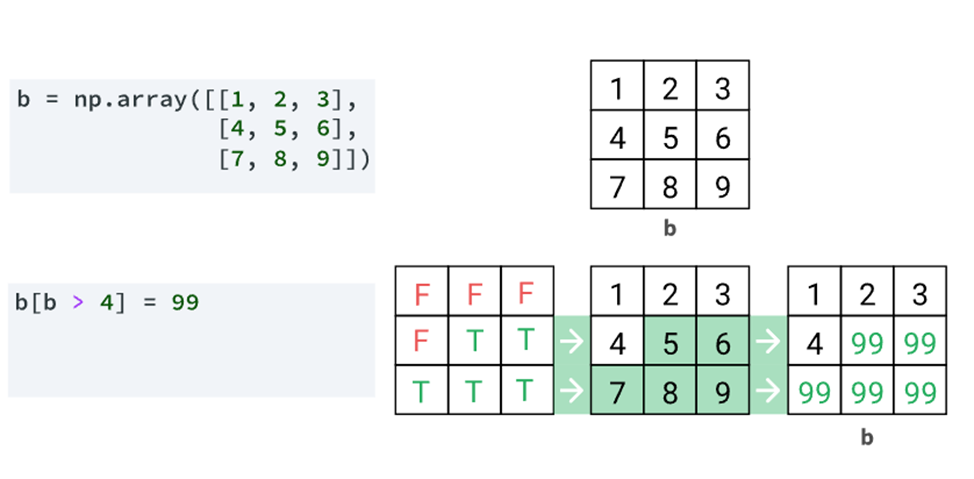


Lưu ý trong sơ đồ trên rằng chúng ta đã sử dụng một "đường tắt" - chèn định nghĩa của mảng Boolean trực tiếp vào vùng chọn. "Đường tắt" này là cách thông thường để viết lập chỉ mục Boolean.

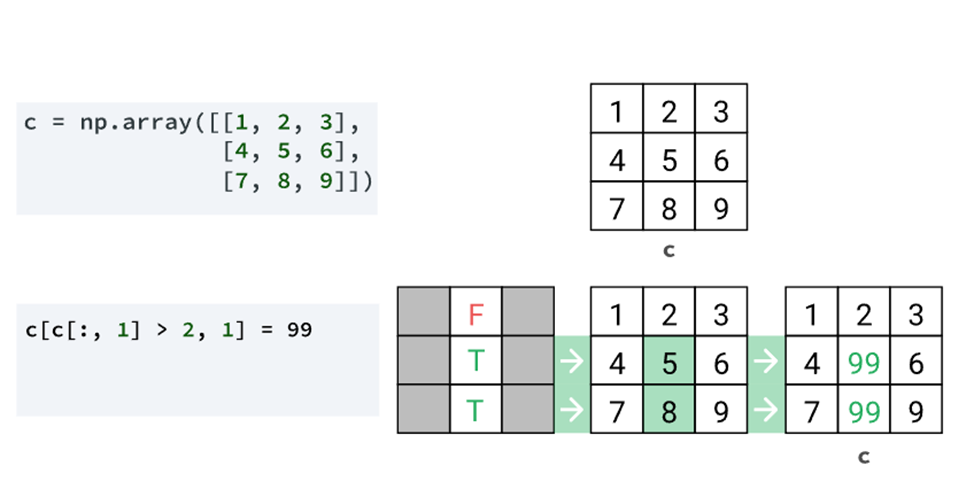
Trong bài tập này, ta lại dùng ndarray.copy() để làm mảng copy

* Lựa cột thứ 14(index 13) trong taxi\_copy, gán kết quả cho total\_amount
* Với hàng mà giá trị của total\_amount nhỏ hơn 0, đổi giá trị đó về 0

Tiếp theo, chúng ta sẽ xem xét một ví dụ về gán bằng cách sử dụng mảng Boolean với hai chiều:



Phép toán b > 4 Boolean tạo ra một mảng Boolean 2D, sau đó kiểm soát các giá trị mà phép gán áp dụng cho.



Phép toán c[:,1] > 2 Boolean chỉ so sánh các giá trị của một cột và tạo ra một mảng Boolean 1D. Sau đó, chúng ta sử dụng mảng Boolean đó làm chỉ mục hàng để gán và 1 làm chỉ mục cột để chỉ định cột thứ hai. Mảng Boolean của chúng ta chỉ áp dụng cho cột thứ hai, trong khi tất cả các giá trị khác không thay đổi

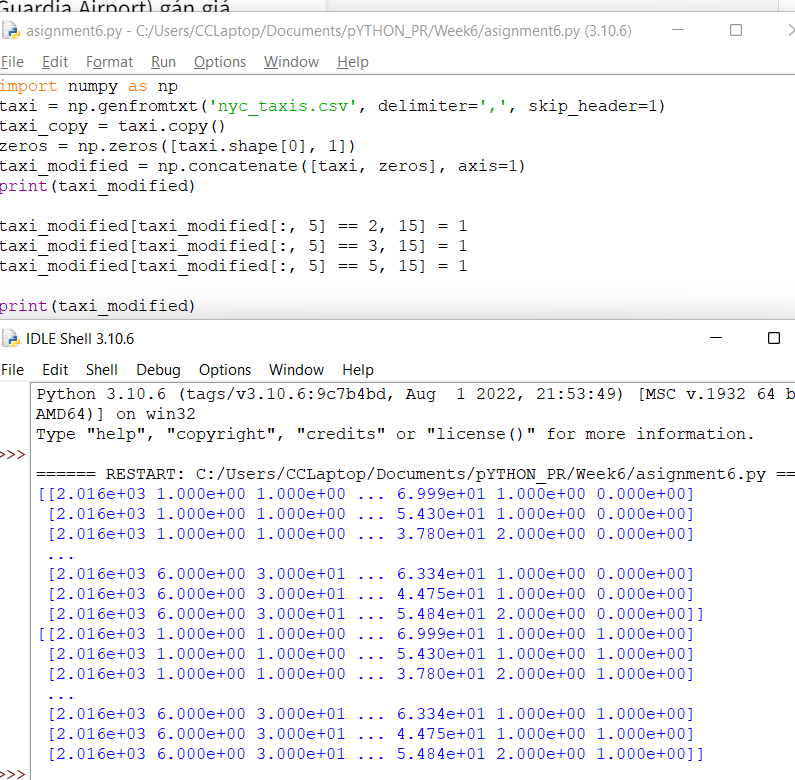
Cú pháp mã giả như sau, đầu tiên sử dụng một biến trung gian

*bool=array[:,collumn\_for\_comparison]==value\_for\_comparision*

*array[bool,column\_for\_assignment]=new\_value*

Tạo một mảng copy từ mảng gốc tên taxi\_modifie, thêm một cột chứa giá trị 0 cho tất cả các hàng

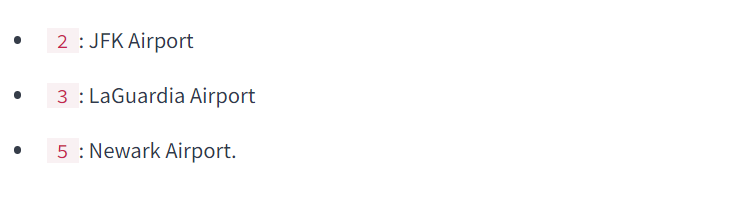
* Với các hàng mà giá trị cột index 5 của nó bằng 2 (JFK Airport) gán giá trị 1 cho cột index 15
* Với hàng mà giá trị cột 5 của nó bằng 3 (LaGuardia Airport) gán giá trị 1 cho cột index 15
* Với hàng mà giá trị cột index 5 bằng 5 thì gán giá trị 1 cho cột index 15



# Bài toán đâu là sân bay được đến nhiều nhất ?

Tìm ra sân bay nào là điểm đến phổ biến nhất trong bộ dữ liệu. Để làm điều đó, chúng ta sẽ sử dụng lập chỉ mục Boolean để tạo ba mảng được lọc và sau đó xem có bao nhiêu hàng trong mỗi mảng

Để hoàn thành tác vụ này, chúng ta sẽ cần kiểm tra xem cột dropoff\_location\_code (chỉ mục cột 6) có bằng một trong các giá trị sau không:



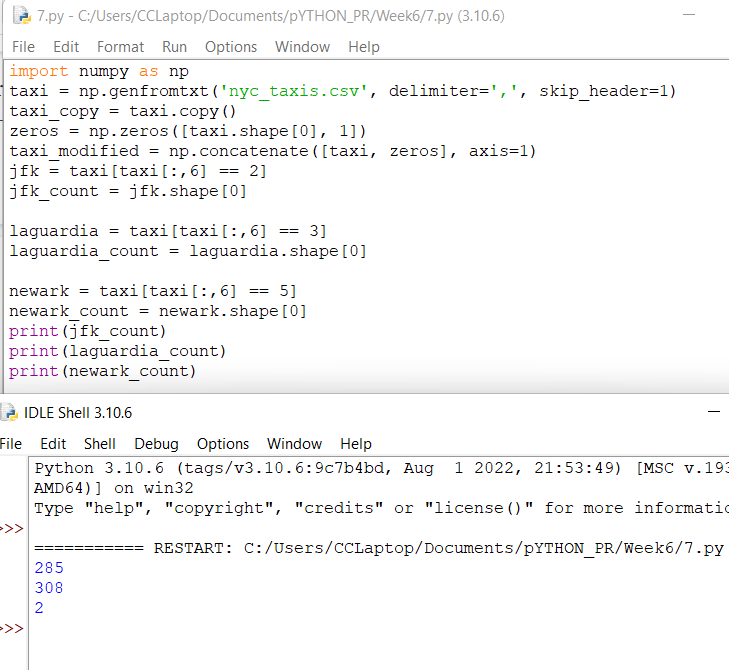
Tính xem có bao nhiêu cuốc xe taxi có mà sân bay LaGuardia là điểm đến

* Dùng phương pháp đánh dấu Boolean để lựa các hàng mà dropoff\_location\_code với cột index 6 của nó có giá trị phản ánh LaGuardia, gán kết quả cho laguardia
* Tính bao nhiêu hàng có trong biến laguardia mới tạo, gán kết quả cho laguardia\_count

Tính xem có bao nhiêu cuốc xe taxi có mà sân bay Newark là điểm đến

* Dùng phương pháp đánh dấu Boolean để lựa các hàng mà dropoff\_location\_code với cột index 6 của nó có giá trị phản ánh LaGuardia, gán kết quả cho newark
* Tính bao nhiêu hàng có trong biến newark mới tạo, gán kết quả cho newark\_count

Sau khi chạy code, quan sát giá trị biến jfk\_count,languardia\_count và newark\_count để xem sân bay nào có nhiều cuốc nhất



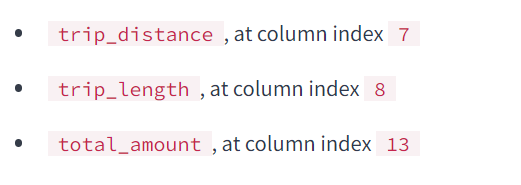
Kết quả xuất ra với LaGuardia là sân bay là địa điểm đến có nhiều taxi nhất: 308

# Bài toán tính dữ liệu các chuyến đi trong một dữ liệu sạch

Tính toán cho thấy LaGuardia là sân bay phổ biến nhất cho các điểm trả khách trong tập dữ liệu này.

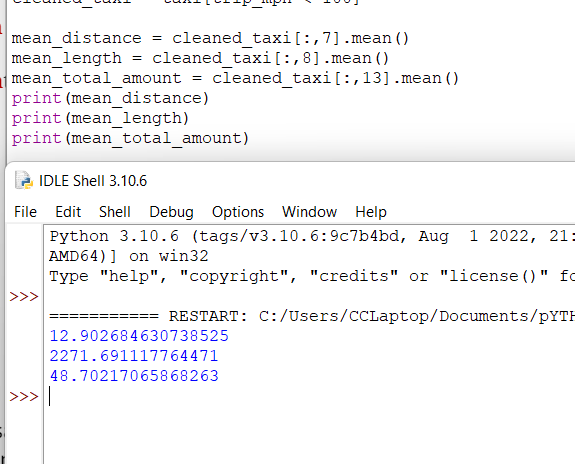
Thách thức cuối cùng liên quan đến việc xóa dữ liệu xấu tiềm ẩn khỏi tập dữ liệu, sau đó tính toán một số thống kê mô tả về dữ liệu "sạch" còn lại

Bắt đầu bằng cách sử dụng lập chỉ mục Boolean để xóa bất kỳ hàng nào có tốc độ trung bình cho chuyến đi lớn hơn 100 dặm / giờ (160 km / giờ) sẽ xóa mọi dữ liệu đáng ngờ. Sau đó, chúng ta sẽ sử dụng các phương thức mảng để tính giá trị trung bình cho các cột cụ thể của dữ liệu còn lại. Dưới đây là các cột đáng quan tâm



Dựa trên các giá trị trip\_mph mà mảng đã cung cấp:

* Tạo mảng mới, cleaned\_taxi, chỉ chứa các hàng mà giá trị trip\_mph của nó nhỏ hơn 100
* Tính toán giá trị mean của trip\_distance của cleaned\_taxi, gán cho mean\_distance
* Tính toán giá trị mean cảu cột trip\_length của cleaned\_taxi, gán cho mean\_lenght
* Tính toán giá trị mean của cột total\_amount của cleaned\_taxi, gán cho mean\_total\_amount



Trong bài học này, chúng ta đã học được những điều sau đây:

* Cách sử dụng numpy.genfromtxt() để đọc trong ndarray
* Giá trị của NaN
* Mảng Boolean là gì và cách tạo một mảng
* Cách sử dụng lập chỉ mục Boolean để lọc các giá trị trong ndarrays một và hai chiều
* Cách gán một hoặc nhiều giá trị mới cho một ndarray dựa trên vị trí của chúng
* Cách gán một hoặc nhiều giá trị mới cho ndarray dựa trên giá trị của chúng