

## 0.1 Optional: print out all of the indices where the maximum value for the successful pick was reached

`np.where` is the method you want; it returns a tuple (think list) with two lists, one for each dimension. To get out the pairs of indices, you want the first element of the first list and the first element of the second list, and so on. You can do this with a for loop, using `zip` to zip together the two arrays.

- for r, c in zip( ):

Some gotchas: it's easiest to do where on the entire `pick_data` set, but then you'll get indices that are NOT the ones you want (other data channels). If you do max on the sliced matrix, then you'll get indices on the sliced matrix, not the original... So either you need to filter out indices that are not the channel you want with an `if` statement (the modulo operator `%` is useful here) OR adjust the column index by doing `offset + index * n_total_dims`.

See the slides for help on converting to and fro from row column indexing, and see the Miro diagram for more info on re-configuring the output of `where`.

This is OPTIONAL for the lab, but we will be coming back to this in the homework.

```
In [93]: # Use where to get out the indices. You can use == OR np.isclose() here; either works. In general,
# floating point comparisons.
# Append the row number of any matches to the max into this list
all_rows_with_max = []
```

```
all_indices_from_where = np.where(np.isclose(pick_data, max_wrist_torque_successful))
# for all row, column in all_indices_from_where
for r, c in zip(all_indices_from_where[0], all_indices_from_where[1]):
    if c // n_time_steps == index_wrist_torque_offset:
        # if this is the column for wrist torque
        print(f"Row: {r}, Time step: {c // n_time_steps} Successful y/n: {pick_data[r, -1] == max_wrist_torque_successful}")
        all_rows_with_max.append(r)

if len(all_rows_with_max) > 0:
    assert all_rows_with_max[0] == 82
else:
    print("No suitable conditions")
```

```
In [92]: grader.check("optional_where")
```

```
Out[92]: optional_where results:
optional_where - 1 result:
Test case failed
```

```

Trying:
    assert all_rows_with_max[0] == 82
Expecting nothing
*****
Line 1, in optional_where 0
Failed example:
    assert all_rows_with_max[0] == 82
Exception raised:
Traceback (most recent call last):
  File "c:\Users\user10\anaconda3\Lib\doctest.py", line 1351, in __run
    exec(compile(example.source, filename, "single",
  File "<doctest optional_where 0[0]>", line 1, in <module>
    assert all_rows_with_max[0] == 82
    ~~~~~^
IndexError: list index out of range

```