

# Solution: Summing and Swapping

This lesson gives a detailed review of the solution to the challenge in the previous lesson.

## We'll cover the following

- Solution
- Explanation

## Solution #

```
def Sum_Swap(df):  
  
    minm_r = np.min(df, axis = 1) # Get minimum elements from all rows  
  
    maxm_r = np.max(df, axis = 1) # Get maximum elements from all rows  
  
    df['row_sum'] = minm_r + maxm_r # Add the min & max values and assign them to new column  
  
    minm_c = np.min(df, axis = 0) # Get minimum elements from all columns  
  
    maxm_c = np.max(df, axis = 0) # Get maximum elements from all columns  
  
    df.loc['col_sum'] = minm_c + maxm_c # Add the min & max values and assign them to new row  
  
    a, b = df['row_sum'].copy(), df.loc['col_sum'].copy() # Store values of row and column in temp  
  
    df['row_sum'], df.loc['col_sum'] = b, a # Interchange the values  
  
    return df  
  
# Test Code  
  
df = pd.DataFrame(np.random.randint(1, 100, 25).reshape(5, 5))  
  
df_res = Sum_Swap(df.copy())  
  
print(df_res)
```



## Explanation #

A function `Sum_Swap` is declared with `df` passed to it as a parameter.

On line 2, the `np.min` function is applied on the `df` to retrieve all the

- On **line 3**, the `np.min` function is applied on the `DataFrame` to retrieve all the minimum values from each row. The `axis=1` indicates that operation should only be performed on the rows. A `Series` is returned containing the minimum values for each row.
- On **line 5**, the `np.max` function is applied on the `DataFrame` to retrieve all the maximum values from each row. The `axis=1` indicates that the operation should only be performed on the rows. A `Series` is returned containing the maximum values for each row.
- On **line 7**, the values of the two `Series` are added and assigned to the newly created column `row_sum`.
- On **line 9**, the `np.min` function is applied on the `DataFrame` to retrieve all the minimum values from each column. The `axis=0` indicates that operation should only be performed on the columns. A `Series` is returned containing the minimum values for each column.
- On **line 11**, the `np.max` function is applied on the `DataFrame` to retrieve all the maximum values from each column. The `axis=0` indicates that operation should only be performed on the columns. A `Series` is returned containing the maximum values for each column.
- On **line 13**, the values of the two `Series` are added and assigned to the newly created row `col_sum`.
- On **line 15**, the values of the column are stored in a temporary variable `a`, and the values of the row are stored in a temporary variable `b`. The `.copy()` creates a deep copy. If this is not used, only their reference is copied, and this is not feasible. It is a good practice to use `.copy()` when dealing with swapping type problems as swapping with references may result in the same value in both variables.
- On **line 17**, the values in the temporary variables are interchanged to the row and column, and hence, the column copy is assigned to the row and the row copy is assigned to the column.
- On **line 19**, the modified `DataFrame` is finally returned.

---

Let's test your understanding of these concepts with a quiz in the next lesson.

