Numpy Practice Questions

1. Define two custom numpy arrays, say A and B. Generate two new numpy arrays by stacking A and B vertically and horizontally.

```
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])

vertical_stack = np.vstack((A, B))
horizontal_stack = np.hstack((A, B))
```

2. Find common elements between A and B. [Hint : Intersection of two sets]

```
A = np.array([1, 2, 3, 4, 5])
B = np.array([3, 4, 5, 6, 7])

common_elements = np.intersect1d(A, B)
```

3. Extract all numbers from A which are within a specific range. eg between 5 and 10. [Hint: np.where() might be useful or boolean masks]

```
A = np.array([1,2,3,4,5,6,7,8,9,10])
A[ np.where((A >= 4) & (A <=9)) ]
```

4. Filter the rows of iris_2d that has petallength (3rd column) > 1.5 and sepallength (1st column) < 5.0

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])
```

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

filter = iris_2d[(iris_2d[:, 2] > 1.5) & (iris_2d[:, 0] < 5.0)]
filter
```

```
array([[4.8, 3.4, 1.6, 0.2],
       [4.8, 3.4, 1.9, 0.2],
       [4.7, 3.2, 1.6, 0.2],
       [4.8, 3.1, 1.6, 0.2],
       [4.9, 2.4, 3.3, 1. ],
       [4.9, 2.5, 4.5, 1.7]])
```

Optional Question:

```
## Optional Practice Question

#Find the mean of a numeric column grouped by a categorical column in a 2D numpy array

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris = np.genfromtxt(url, delimiter=',', dtype='object')
names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

numeric_columns = iris[:, 0:4].astype('float')  # sepalwidth
grouping_column = iris[:, 4]  # species

unique_species = np.unique(grouping_column)

output = []

# Calculate the mean for each unique category
for species in unique_species:
    category_values = numeric_columns[grouping_column == species]
    category_mean = np.mean(category_values)
    output.append((species, category_mean))

print(output)
```

```
[(b'Iris-setosa', 2.533), (b'Iris-versicolor', 3.573000000000001), (b'Iris-virginica', 4.285)]
```

Practice Questions for Pandas

1. From df filter the 'Manufacturer', 'Model' and 'Type' for every 20th row starting from 1st (row 0).

```
df = pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv')
```

```
filtered_data = df [ [ 'Manufacturer', 'Model', 'Type' ] ] . iloc [ :: 20]
```

2. Replace missing values in Min.Price and Max.Price columns with their respective mean.

```
df = pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv')
```

```
min_price_mean = df [ 'Min.Price' ] .mean()
max_price_mean = df [ 'Max.Price' ] .mean()

df['Min.Price'].fillna(min_price_mean, inplace=True)
df['Max.Price'].fillna(max_price_mean, inplace=True)
```

3. How to get the rows of a dataframe with row sum > 100?

```
df = pd.DataFrame(np.random.randint(10, 40, 60).reshape(-1, 4))
```

```
row_sums = df .sum (axis = 1)
result = df [ row_sums > 100 ]
```